Diplomarbeit vom 6. September bis 29. Oktober 2001

Studenten: José Fontanil und Reto Glanzmann

Dozent: Patrick Feisthammel

PhPepperShop

José Fontanil / Reto Glanzmann November 2001

Inhaltsverzeichnis

1. Bericht

1.1 Kurzbeschrieb in Deutsch und Englisch....3
1.2 Vorwort....5
1.3 Aufgabenstellung....6
1.4 Analyse des erhaltenen Feedbacks....8
1.5 Wahl der verwendeten Technologien....12
1.6 Zeitplanung....14
1.7 Tests....18
1.8 Erfahrungen....24
1.9 Zukunft des PhPepperShops....60
1.10 Schlusswort....62

2. Architektur

2.1 Security....63
2.2 PHP-Module und -Funktionen....66
2.3 Session Management....100
2.4 Datenbank....102
2.5 Shop Layout Management....111
2.6 Artikelmanagement....117
2.7 Kategorienmanagement....121
2.8 Zahlungsmöglichkeiten....125
2.9 Bestellungsmanagement....128
2.10 Kundenattribute Management....129
2.11 CRM – Customer Relationship Management....130

3. Manuals

3.1 Installation / Deinstallation....132
3.2 Backup und Restore mit MySQL....135
3.3 Bedienung....139

4. Anhang

4.1 Literaturverzeichnis....149
4.2 Software Versionen....152
4.3 Projekttagebuch....153
4.4 GPL – GNU General Public Licence....158
4.5 Detailliertes Inhaltsverzeichnis....161
4.6 Inhalt der CD....166
4.7 Zusammenfassungen....167
4.8 Source Code....168

 © 2001 R. Glanzmann & J. Fontanil

1. Bericht

1.1 Kurzbeschrieb in Deutsch und Englisch

1.1.1 Kurzbeschrieb (Deutsch)

Die Entwicklung eines frei verfügbaren, beliebig konfigurier- und erweiterbaren Open Source Webshops war das Ziel, welches wir uns zu Beginn unserer Diplomarbeit setzten.

Entstanden ist ein Shopsystem, welches Artikel mit beliebig vielen Optionen und Varianten unterstützt. Für jeden Artikel kann ein beliebig grosses Artikelbild gespeichert werden, von welchem automatisch ein Thumbnail (kleines Vorschaubild mit fest definierter Grösse) erzeugt wird. Der Shopkunde kann sich bei Bedarf seine Adressdaten benutzername- und passwortgeschützt speichern lassen, damit bei einem späteren Einkauf die Adresse nicht nochmals eingegeben werden muss. Per Suchfunktion kann komfortabel im Shop nach bestimmten Artikeln gesucht werden. Verschiedene frei konfigurierbare Zahlungsarten sind implementiert, so dass mittels Payment-Schnittstelle mit wenig Aufwand die Anbindung an ein Online-Payment Institut möglich ist.

Der Shopadministrator tätigt alle Einstellungen direkt per Webinterface. Ein umfangreiches Layout-Management ermöglicht es, das Shopaussehen bis ins letzte Detail zu konfigurieren (Hintergrundbilder, Framegrösse, Shoplogo, eigene Buttons). Artikel können erstellt, verändert oder gelöscht werden. Ausserdem ist es möglich, einen Artikel mehreren Kategorien zuzuordnen. Per Kategorienmanagement können Kategorien und Unterkategorien erstellt, umbenannt, verschoben und gelöscht werden. Optional kann für das Entgegennehmen der Kreditkartendaten eine SSL-Unterstützung aktiviert werden. Versandkosten werden nach Gewicht, Preis oder Pauschale verrechnet. Ein Mindermengenzuschlag und Nachnahmegebühren können dynamisch zum Rechnungstotal addiert werden.

Der Shop wurde in den Scriptsprachen PHP und JavaScript entwickelt. Die HTML-Seiten werden dynamisch per PHP erzeugt und Benutzereingaben mittels JavaScript auf ihre Korrektheit überprüft. Alle Daten werden in einer MySQL-Datenbank gespeichert. Die Installation erfolgt bequem mittels eines Perl-Scripts.

Wie schon im Titel erwähnt, wurde der Shop als Open Source Projekt auf SourceForge entwickelt. Das heisst, dass der Quellcode frei verfügbar ist und der Shop uneingeschränkt verwendet und weiterentwickelt werden kann. Aktuelle Informationen und einen Beispielshop finden sich auf der Seite http://www.phpeppershop.com.

1.1.2 Abstract (English)

Our aim was to develop a web shop that could be configured flexibly, distributed freely and is open source under the GNU-General Public Licence. What has arisen is a shopping system that offers articles with infinite options and variations.

A picture of a desired size can be assigned to each article. Due to an automatically generated thumbnail of the before mentioned picture, the customer is given a consistent view of the articles.

The customer can optionally save his address. He can then access his record by simply logging

 © 2001 R. Glanzmann & J. Fontanil

in with a login name and password. This way he doesn't have to enter his address more than once. The shop provides comfortable search functions to find articles. Several types of payment are supported: cash before delivery, cash on delivery, on account and credit card payment. Thanks to a flexible payment-interface it is possible to link the PhPepperShop to an online-payment institute with ease. PhPepperShop can optionally encrypt the credit card input via SSL.

The shop administrator can access the web shop remotely from any location via a web interface. A flexible layout allows for changing the look of the shop in detail (background images, frame size, shop logo, buttons, etc.). Articles can be created, updated and deleted. An article can even be assigned to more than one category. With the category management the shop administrator can easily create, rename, replace and delete categories and sub categories. Shipping costs can be charged by price, by weight or by an estimated amount. Small amount surcharge and pay on delivery surcharge can be added to the invoice automatically.

The web shop has been programmed using PHP and JavaScript. The HTML pages are generated by PHP, JavaScript validates customer input. The whole shop is saved in a MySQL database. The installation is unlike UNIX with an automatic installation tool written in PERL.

As mentioned in the title, the PhPepperShop is an open source project, which includes free availability of the source code and unrestricted utilisation and development. The PhPepperShop project is maintained on Source Forge. The latest news and a sample shop can be found on www.phpeppershop.com.

 © 2001 R. Glanzmann & J. Fontanil

1.2 Vorwort

Im Rahmen unserer Diplomarbeit erweiterten wir unsere erste Projektarbeit "Webshop Framework in PHP" zu einem marktreifen Produkt, dem PhPepperShop.

Das wichtigste Ziel für uns war, den Shop soweit zu entwickeln, dass man ihn ohne Bedenken der Öffentlichkeit übergeben kann. Wir wollten den Shop so programmieren, dass er möglichst vielseitig und branchenunabhängig eingesetzt werden kann. Weiter sollte sich der Shop komplett auf die Bedürfnisse des Shopbetreibers anpassen lassen. Dabei berücksichtigten wir auch die Wünsche und Verbesserungsvorschläge des inzwischen bei uns eingegangenen Feedbacks.

Ein von Anfang an erklärtes Ziel war es, den Shop frei von lizenzrechtlichen Einschränkungen auf einem öffentlich zugänglichen Server gehostet anzubieten. Wir vertreten mit voller Überzeugung die Idee der Open Source Software Entwicklung.

 © 2001 R. Glanzmann & J. Fontanil

1.3 Aufgabenstellung

Diplomarbeit: Open Source Webshop "PhPepperShop"

J. Fontanil
R. Glanzmann

Umfeld

Ziel dieser Diplomarbeit ist die Erweiterung des in einer Projektarbeit erstellten *Framework für einen Webshop in PHP*.

Der Shop soll funktionell soweit fertiggestellt werden, dass der praktische Einsatz möglich ist. Dazu soll der Shop als Open Source Projekt publiziert werden und wenn möglich eine *echte (praktisch eingesetzte)* Installation erfolgen. Die daraus gewonnenen Erfahrungen sollen umgesetzt werden.

Aufgaben

Im Umfang der Diplomarbeit sind folgende Punkte zu erfüllen:

- Erstellen des WebShops gemäss *Aufgabendetails*
- Erstellen der Dokumentation

Design-Entscheidungen sind zu dokumentieren und zu begründen.

Die Dokumentation hat zudem mindestens zu umfassen:

- Architektur (Komponenten und deren Funktionalität, sowie Workflow)
- Projekttagebuch (Zeitplanung, Probleme, Entscheidungen, Erfahrungen)
- Ausblick (Verbesserungsmöglichkeiten, Lehren)

Die im Rahmen der Publizierung veröffentlichten Dokumente sind Bestandteil der Dokumentation.

Aufgabendetails

Es geht in dieser Arbeit darum den WebShop zu ergänzen mit:

- Custom Relationship Management (CRM)
- Layout Management mit CSS
- Online Payment
- Implementation der wichtigsten Zusatzfunktionen gemäss Besprechung zu Beginn der Diplomarbeit
- Fertigstellung des Administrationstools

Zudem sollen Erfahrungen mit der Publikation eines Open Source Projekts auf SourceForge gesammelt und dokumentiert werden.

Falls genügend Zeit vorhanden ist, sind weitere Ergänzungen denkbar:

- Automatische Katalogerzeugung im PDF Format
- Offline Administration
- Erweiterung der Funktionalität

 © 2001 R. Glanzmann & J. Fontanil

Termine

Donnerstag, 6. September 2001	Ausgabe der Diplomarbeit
Montag, 29. Oktober 2001	Abgabe der Arbeit
Montag, 12. November 2001	Präsentation der Diplomarbeit
Donnerstag, 22. November 2001	Industriepräsentation
Samstag, 24. November 2001	Umzug, Ausstellung

Kontaktmöglichkeiten

Ihr Betreuer ist per Telefon und E-Mail erreichbar. Die Details werden in der ersten Besprechung angegeben.

Abgabe

Alle Daten (Dokumente, Programmcode, etc.) sind vollständig elektronisch abzugeben. Benötigte Zusatzsoftware (falls lizenzrechtlich erlaubt) ist ebenfalls elektronisch abzugeben.

Die Berichte sind zusätzlich in gedruckter Form abzugeben.

Bewertung

Die Bewertung der Arbeit erfolgt mit folgendem Bewertungsschema:

Kriterium	**Gewicht**
Fleiss, Einsatz, Interesse	1
Methodik, Vorgehen bei der Problemlösung	3
Eigeninitiative, Kreativität, Originalität	4
Gründlichkeit, theoretische Betrachtungen	3
Erarbeitetes Wissen, Literatur, Vollständigkeit	3
Implementation	
Techn. Korrektheit, Bedienbarkeit der Programme	2
Dokumentation, Verständlichkeit Programmcode	2
Vortrag	
Eingehen auf das Wesentliche, Einsatz der Medien	2
Verständlichkeit, Einhalten der Zeit	2
Bericht	
Gliederung, Darstellung	3
Technische Korrektheit, Referenzen	3
Umfang, Vollständigkeit, Sprache	2

Näheres zum Projekt findet man auf: http://www.phpeppershop.com
http://www.zhwin.ch/~fei/da/

 © 2001 R. Glanzmann & J. Fontanil

1.4 Analyse des erhaltenen Feedbacks

Unsere Diplomarbeit setzt auf der Projektarbeit 'Framework für einen Webshop in PHP' (PA Fei01/2) auf. Wir veröffentlichten die Arbeit am 19. Juni 2001 und haben seither über 15'000 Hits auf dieser Homepage aus den verschiedensten Ländern verzeichnet. Da ist es klar, dass uns auch einige Leute E-Mails geschrieben haben. Weiter haben wir Feedbacks von mehreren freiwilligen Shop-Testern erhalten. Zudem gaben uns auch Shopbetreiber Tipps.

1.4.1 E-Mail Auswertung

Anzahl	*Vorschlag*
7	Englische Sprache erwünscht (allgemein Mehrsprachensupport)
1	Englischer Code erwünscht (vor allem Kommentare)
9	Provider Probleme ? 1 User, (ev. auch nur eine Datenbank), detaillierte Installationsanleitung
1	htaccess erklären
2	SSL allgemein, und vor allem bei Formular Übertragung zur Bank / yellowpay / SECPay.com
1	Suchfunktionen auf der Homepage
1	Stern-Zeichen als Admin-Login wegtun ? ersetzen durch Admin-Entry-Page
5	Windows Version des Shops (nur NT / 2000)
1	Mehrwertsteuer und Verpackungs-/Frachtkosten in die Berechnung miteinbeziehen
2	Lagerverwaltung: Bestand eingeben, bei Verkauf dekrementieren -> Mindermengen Alarm
1	Verfügbarkeit des Artikels angeben und ev. Lieferterminangaben
1	Dynamisches Page Layout (auch: Header und Footer)
2	Unterkategorien (z.B. 300 Kategorien und mehr Produkte)
3	CRM, personalisierte Artikelliste, eigenes Login, autom. ausgefüllte Adresse (weiter: persönlicher Ansprechspartner konfigurieren können, Rechnung, Lieferschein drucken können, Bestellungsfortschritt abrufbar(versendet, ...))
1	Shop muss mit root-Rechten laufen (mühsam, wegen Bild Upload)
1	Developer-Mailing Liste
1	CVS-Server
3	Gleicher Artikel mit verschiedenen Variationen/Optionen können nicht kumuliert werden
1	Optionen/Variationen als Pop-Up Menü implementieren (ev. nur optional)

Weitere Tipps aus den E-Mails

- Wir sollten Webmastern den Shop empfehlen (Marketing)
- Artikel in der c't oder ähnlichen Magazinen anstreben

 © 2001 R. Glanzmann & J. Fontanil

- Wir sollten das Problem der Secure-Online-Transactions genau anschauen (P. Fitzpatrick)

1.4.2 Auflistung aller erhaltenen Feedbacks

Wir haben folgende Liste mit möglichen Arbeiten aus allem erhaltenen Feedback zusammengestellt und im Rahmen der Zeitplanung (Kapitel 1.6) einer Priorisierung und Zeitaufwandschätzung unterzogen.

Arbeitsvorschläge
Shopname suchen und definieren
Shop Administrationsmodus fertig stellen
Formular Buttons durch grafische Buttons ersetzen
Benutzerinfo, dass ohne E-Mail Adresse keine Bestellbestätigung ausgestellt wird
Admin soll entscheiden können ob der Warenkorb direkt angezeigt werden soll
Aktionspreis implementieren
Artikel in mehreren Kategorien eintragen können
Anleitung für GIF / PNG Bilder Benutzung (für Artikel Bilder)
Warenkorb ist leer Meldung implementieren
Button hinzufügen, der mehr leere Options- / Variationsfelder 'generiert'
Kreditkartenunterstützung ? ev. noch Wrapperklasse für mehrere Payment Institute (Schnittstelle)
Importmöglichkeit (von anderen Quellen Artikeldaten importieren können)
Kategorienmanagement
Offline Administrationstool
Suchresultate optional mit Bilder
Anzahl Artikel pro Suchresultat-Seite festlegen (z.B. 25 Artikel pro Seite), Kunde kann wählen: 10 ... 50 ... alle
Suchanfrage mit mehreren Begriffen (à la Google®, AND-verknüpft)
Eigene Buttons (Buttonsets) ev. mit JavaScript (onmouseover, onclick,...)
Felderkonfiguration bei Kundenadresseingabe dynamisch wählbar gestalten (Admin) ? Reihenfolge wählbar
SSL ? GNU-PG ? E-Mail (Kreditkarten Nr., ...) Mailorder-Vertrag, allgemein SSL anschauen
Kostenlose Artikel ? z.B. Prospekte bestellen ? Zahlungsmöglichkeiten weglassen können
PLZ kann auch alphanummerisch sein (USA, oder mit voran gestelltem Ländercode)
Backup des Shops (ev. auch automatisiert ? immerhin Anleitung)
Charset im Mailversand ergänzen
Englische Sprache erwünscht (allgemein Mehrsprachensupport)

 © 2001 R. Glanzmann & J. Fontanil

Englischer Code erwünscht (v.a. Kommentare)
Provider Probleme ? 1 User, (ev. 1 Datenbank), detaillierte Installationsanleitung
htaccess erklären
Suchfunktionen auf der Homepage
Sternchenpixel als Admin-Login wegtun ? ersetzen durch Admin-Entry-Page
Windows (NT / 2000) Version des Shops ? Dozent nach Erfahrung fragen ? Links weg tun
Mehrwertsteuer und Verpackungs-/Frachtkosten in die Berechnung miteinbeziehen
Lagerverwaltung: Bestand eingeben, bei Verkauf dekrementieren -> Mindestmengen Alarm
Verfügbarkeit des Artikels angeben und ev. Lieferterminangaben
Dynamisches Page Layout --> CSS automatisch erzeugen lassen
Unterkategorien erlauben
CRM 1:Eigenes Login, automatisch ausgefüllte Adresse, Login zwingend/nicht zwingend ? Händlermodus
CRM 2: weiter: persönlicher Ansprechpartner konfigurieren können, Rechnung, Lieferschein drucken können, Bestellungsfortschritt abrufbar (versendet, ...)
Shop muss mit root-Rechten laufen (mühsam, wegen Bild Upload) ? Problem lösen
SourceForge: CVS-(-Server), Developer-Mailing Liste, ...
Gleicher Artikel mit verschiedenen Variationen/Optionen können nicht kumuliert werden ? Fehler beheben
Variationen als optionales Pulldown-Menü implementieren
Scrolling im Top-Frame unterbinden
Shop-Logo upload ermöglichen ? Als (GIF-)Bilddatei im Bilder-Verzeichnis ablegen
Bilddatum für Cache auf 1.1.1971 korrigieren
Authentifizierung anstatt über htaccess über die PHPLib (PHPAuth)
Neuprogrammierung der Artikel-Update Funktion(-en)
Mehrere Shops in einer DB (Präfix bei Scripts u.s.w.)

1.4.3 Erste Vorentscheidungen

Wir haben uns nach einer groben Aufwandschätzung für das Weglassen von verschiedenen Punkten entscheiden müssen. Gegen die Ausführung der Arbeiten haben wir uns entschieden wenn:

- Die Arbeit als reine Fleissarbeit, ohne Lerneffekt angesehen wurde (z.B. Übersetzung des Shops, Windows Portierung).
- Die Implementierung eines funktionsfähigen, einsatzbereiten Shops aufgrund des Aufwands in Frage gestellt werden musste (z.B. CRM 2).

Uns war vor allem wichtig, dass der Shop im praktischen Einsatz bestehen kann. Versand-

 © 2001 R. Glanzmann & J. Fontanil

kosten zum Beispiel waren einfach ein Muss, denn ohne Versandkostenrechnung kann man nicht wirklich einen Shop betreiben. Erst danach haben wir uns für Zusatzfunktionen wie eine erweiterte Artikelsuche entschieden.

Sind wir ein Problem angegangen, so entschieden wir uns wenn immer möglich dafür, in die jeweilige Teilarbeit auch mehrere Überlegungen einfliessen zu lassen. Wir mussten zum Beispiel das Problem lösen, dass mehrere PhPepperShops gleichzeitig in einem DBMS laufen können. Weiter sollte die Installation besser dokumentiert werden, den 1-Datenbank-User Betrieb implementiert und die Möglichkeit geschaffen werden, mit schon bestehenden Datenbank-User arbeiten zu können. Wir haben alle diese Einzelprobleme zusammen gefasst und eine automatisierte Installation erstellt. Auf diese Weise sind alle Probleme elegant gelöst und dem Shopadministrator wird viel mühsame, fehlerträchtige Arbeit abgenommen.

Welche Arbeiten schliesslich ausgewählt wurden, sieht man im Kapitel 1.6, Zeitplanung. Dort ist auch ersichtlich, wie wir den Aufwand der einzelnen Arbeiten eingeschätzt haben.

 © 2001 R. Glanzmann & J. Fontanil

1.5 Wahl der verwendeten Technologien

Ein Framework für einen Webshop lässt sich grundsätzlich auf viele Arten realisieren. In den folgenden Absätzen wollen wir begründen, wieso wir uns für die Kombination PHP4 als Apache-Modul und MySQL als Datenbank entschieden haben.

Ein vielfältiger Webshop sollte ein hohes Mass an Dynamik bieten, sowohl vom Workflow her, als auch innerhalb der HTML-Seite. Weiter sollte er seine Daten, wenn möglich, aus einer zentralen Datenbank beziehen können. Damit hat man auch alle vier Komponenten, die nötig sind um einen dynamischen Webshop realisieren zu können:

- Clientseitige Scriptsprache um statische HTML Seiten dynamisch zu verändern
- Serverseitige Scriptsprache zur Erstellung dynamischer Websites
- Webserver
- Datenbank

Um statischen HTML-Seiten Dynamik zu verleihen fällt die Wahl fast automatisch auf JavaScript. Diese umfassende Scriptsprache erlaubt es, eine bereits komplett herunter geladene Webpage dynamisch zu verändern. Man kann mit JavaScript auch einen Teil der Funktionalität, welche man serverseitig ausführen müsste, auf den Client verlegen, um so die Rechenlast auf dem Server zu minimieren. Es war uns bewusst, dass JavaScript viele Probleme mit sich bringen würde. Wir wollten aber auf die Dynamik, die damit gewonnen wird, nicht verzichten.

Da wir uns für eine Veröffentlichung des Webshops unter der GPL (siehe Anhang) entschieden haben, war es uns auch wichtig Technologien zu verwenden, die selbst Open Source sind. Deshalb wählten wir PHP4 (Personal Home Page Tools Version 4). Unter anderem weil PHP fürs Web erstellt wurde und nicht wie z.B. ASP, das mit VBScript daher kommt, welches nur 'schwer' an die Bedürfnisse eines Webservers angepasst werden kann und ausschliesslich für die Microsoft Plattform verfügbar ist. Perl hingegen bietet sehr mächtige Zeichenketten-Funktionen, hat aber den Nachteil einen eher unleserlichen Code zu eigen zu haben. PHP nimmt sich der Nachteile der oben genannten Technologien an und verbindet diese mit ihren Vorteilen. PHP bietet die von ASP bekannte, einfache Kombination mit HTML, die leistungsstarke und weit verbreitete C-Syntax, die vielen schon von JAVA und JavaScript her bekannt ist. PHP besticht ausserdem durch einen fast schon grandiosen Funktionsumfang. Hier kann keine andere Sprache mithalten. Das sieht man vor allem bei der Datenbank Unterstützung, die uns auch sehr wichtig war.

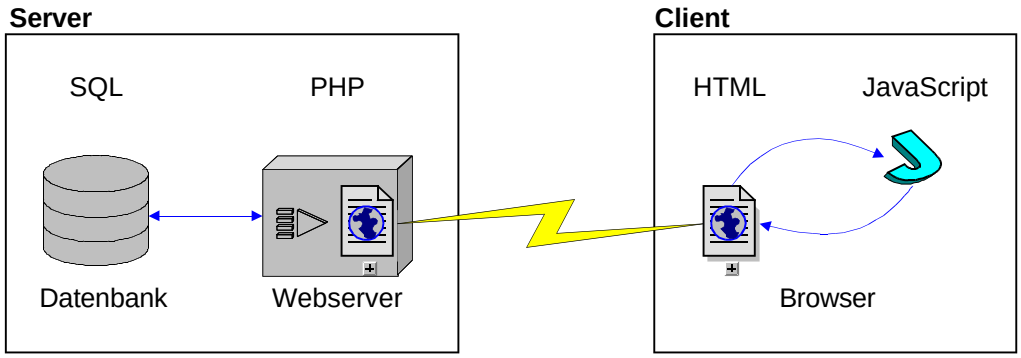
Die Entscheidung welchen Webserver wir verwenden sollten, war einfach zu treffen: Apache ist der heute im Internet am meisten eingesetzte Webserver: zuverlässig, Open Source und für die wichtigsten Plattformen verfügbar. Er hat auch grosse Zukunftsaussichten. Wir wollten den Einsatz des Shops aber nicht grundsätzlich an eine Webserver-Technologie binden. Es braucht nur minimalste Änderungen, um den Shop auf einem anderen Webserver laufen zu lassen (Passwortschutz für Administration und Pfade bei Wechsel von UNIX zur Windows-Plattform).

Bei der Datenbank konnten wir eine gute Idee unseres Dozenten und Diplomarbeitbetreuers, Herr Feisthammel, umsetzen. Unser Webshop wickelt alle Datenbankzugriffe zentral über einen Datenbank-Wrapper ab. Das heisst, der Shop-Betreiber kann wählen, welche Datenbank er benutzen will. Dies ist sicher von Vorteil, wenn in einem bestehenden Unternehmen schon eine Datenbank vorhanden ist. Der Shop ist momentan zu 100% auf MySQL einsetzbar und getestet. Die Programmierung der Sybase Unterstützung ist so gut wie abgeschlossen,

 © 2001 R. Glanzmann & J. Fontanil

für alle weiteren Datenbanken muss man nur noch zwei Klassen neu schreiben.
MySQL war für uns erste Wahl, weil diese Datenbank für jedermann gratis zur Verfügung steht und bei den Internet Service Providern vielfach schon vorinstalliert ist. Ihr Leistungsumfang kommt zwar nicht an den grosser Datenbanken heran, wir haben uns aber mit dem DB-Wrapper diese Türe offen gehalten.

Abschliessend sei hier noch visuell dargestellt, mit welcher Technologie die Teile des PhPepperShops realisiert wurden. Die Datenbank wird über PHP via SQL Statements angesprochen. PHP generiert danach dynamisch HTML Code. Die HTML-Seiten werden über das Internet an den Browser übertragen. Clientseitig kann ein HTML Dokument per JavaScript dynamisch gestaltet werden. Was im folgenden Bild nicht erwähnt wird, ist das in Perl und Shell-Script geschriebene Installations- und Deinstallations-Script.



 © 2001 R. Glanzmann & J. Fontanil

1.6 Zeitplanung

Nach der Auswertung der Feedbacks und Anregungen, die wir zwischen der ersten Projektarbeit und der Diplomarbeit erhalten hatten, stellten wir eine Liste mit Tätigkeiten und zu implementierenden Funktionalitäten auf. Wir teilten sie in Gruppen unterschiedlicher Dringlichkeit und Wichtigkeit ein. Ausserdem ordneten wir die Tätigkeiten verschiedenen Tätigkeitsgruppen zu. Des weiteren nahmen wir für jeden zu erledigenden Punkt eine grobe Aufwandabschätzung vor. Aufgrund dieser ermittelten Daten wollten wir eine erste detaillierte Zeitplanung erstellen. Die Zeitaufwandsumme der dringend zu erledigenden Arbeiten überstieg jedoch schon fast den gesamten Zeitraum, den wir zur Programmierung zur Verfügung hatten.

Da es unser allerhöchstes Ziel war, den Shop nach Ende der Diplomarbeit funktionsfähig der Öffentlichkeit zu übergeben und wir keine Abstriche in der Funktionalität machen konnten, erstellten wir eine überaus ehrgeizige Zeitplanung. Wir entschieden uns, absolut strikt und ohne Ausnahme nach der Prioritätenliste vorzugehen und Versuche sowie Exkurse weitgehendst zu unterlassen. Da wir unseren Shop auf dem CVS-Server von SourceForge entwickelten, hatten wir fast täglich wieder Commits (hochladen von aktualisierten oder neu erstellten Softwareteilen auf den CVS-Server) und Checkouts (herunterladen der aktuellen Version des Programms). Schnell merkten wir jedoch, dass die nahezu tägliche Neuinstallation des Shops viel Zeit auffrass. So entschlossen wir uns kurzfristig ein Installations- und Deinstallationstool zu erstellen, was rückblickend gesehen eine sehr lohnende Entscheidung war, da wir uns dadurch viel Installationsaufwand und Installationsfehler ersparen konnten.

Durch den Einsatz sämtlicher uns zur Verfügung stehender Kräfte, kamen wir dann bei den meisten Punkten schneller voran, als wir dies eigentlich ursprünglich eingeplant hatten. Wobei hier erwähnt werden muss, dass wir bei der ursprünglichen Planung von einer durchschnittlichen Arbeitsstundenanzahl von 8½ Stunden pro Tag und Person ausgingen, welche wir schlussendlich jedoch meistens um mehrere Stunden überschritten. Ein mühsamer Knackpunkt war das Layoutmanagement, da wir massiv mit der unterschiedlichen Interpretation von HTML, der von uns zu Testzwecken eingesetzten Browser, zu kämpfen hatten.

Gerne hätten wir mehr Zeit in die Zahlungsabwicklung mit Online-Payment-Instituten investiert. Nach einem Besuch dieser Firmen an der Orbit/Comdex, mussten wir jedoch feststellen, dass sich die Zielgruppe dieser Payment-Anbieter nicht mit der unseres Shops deckt. Wir planten dann ein wenig um und entschlossen uns, selbst eine sichere Zahlungsabwicklung zu implementieren.

Erschwerend zur ohnehin schon knapp kalkulierten Zeitplanung kam dazu, dass vor allem in der Start- und Endphase der Diplomarbeit das ZHW-LAN und die Verbindung zum Internet oft und über längere Zeit nicht verfügbar waren, was bei einem Webprojekt nicht gerade förderlich ist.

1.6.1 Aufgabenauflistung nach Wichtigkeit, 17.09.2001

Legende:

Priorität (P):	1 = höchste 12 = kleinste	**Zeitaufwand (Z):**	1 = bis halber Tag 2 = bis zwei Tage 3 = mehr Zeit wird benötigt	**Farben:**	wichtigste Aufgaben wichtige Aufgaben nicht so wichtige Aufgaben nice to have

© 2001 R. Glanzmann & J. Fontanil

Bekannte Fehler und Mängel beheben

P	Z	*Aufgabe*
1	1	Provider Probleme --> 1 User, (ev. 1 Datenbank), detaillierte Installationsangaben
2	3	Gleicher Artikel mit verschiedenen Var. / Optionen können nicht kumuliert werden
3	3	Unterkategorien (z.B. 300 Kategorien und mehr Produkte)
4	3	Abändern der Artikel-Update Funktionen
5	3	Artikel in mehreren Kategorien eintragen können
6	3	Mehrwertsteuer und Verpackungskosten/Frachtkosten in die Berechnung miteinbeziehen
7	1	Scrollen im Top-Frame unterbinden
8	1	Charset im Mailversand ergänzen
9	1	Bilddatum für Cache auf 1.1.1971 korrigieren
10	1	Kostenlose Artikel --> z.B. Prospekte bestellen --> Zahlungsmöglichkeiten weglassen,..
11	1	PLZ auch alphanumerisch (USA)
12	2	Shop muss mit root-Rechten laufen (mühsam, wegen Bild Upload)

Admin-Modus fertig stellen

1	3	Shop-Administrationsmodus fertig stellen (Zahlungsmöglichkeiten, Kategorienmanagement, Unterkategorien, u.s.w.)
2	3	Kategorienmanagement
3	1	Button hinzufügen, der mehr leere Options/Variantenfelder 'generiert'
4	2	Felderkonfiguration bei Kundenadresseingabe dynamisch wählbar gestalten (Admin) --> Reihenfolge wählbar
5	2	Aktionspreis implementieren
6	1	Formularbuttons durch grafische Buttons ersetzen (überall!)

Implementieren wichtiger Zusatzfunktionalität

1	3	Kreditkartenunterstützung --> ev. Wrapperklasse für mehrere Institute
2	1	Warenkorb ist leer Meldung implementieren
3	2	Backup des Shops (ev. auch automatisiert --> wenigstens Anleitung)
4	1	htaccess erklären
5	3	Anzahl Artikel pro Suchresultatseite festlegen (z.B. 25 Artikel pro Seite), Kunde kann wählen: 10 ... 50 ... alle
6	2	Suchanfrage mit mehreren Begriffen (wie bei http://www.google.com)
7	1	Administrator soll entscheiden können ob Warenkorb direkt angezeigt werden soll
8	1	Anleitung für GIF/PNG Bilder Benutzung (für Artikelfotos)
9	1	Benutzerinfo, dass ohne E-Mail Adresse keine Bestellbestätigung ausgestellt wird
10	3	SSL --> GNU GP <-- PGP-Mail (Kreditkarten Nr., ...) Mailorder-Vertrag / allgemein SSL anschauen --> wo einsetzen
11	2	Suchresultate optional mit Bilder
12	3	Lagerverwaltung: Bestand eingeben, bei Verkauf dekrementieren --> Mindestmengen Alarm

Layout Management

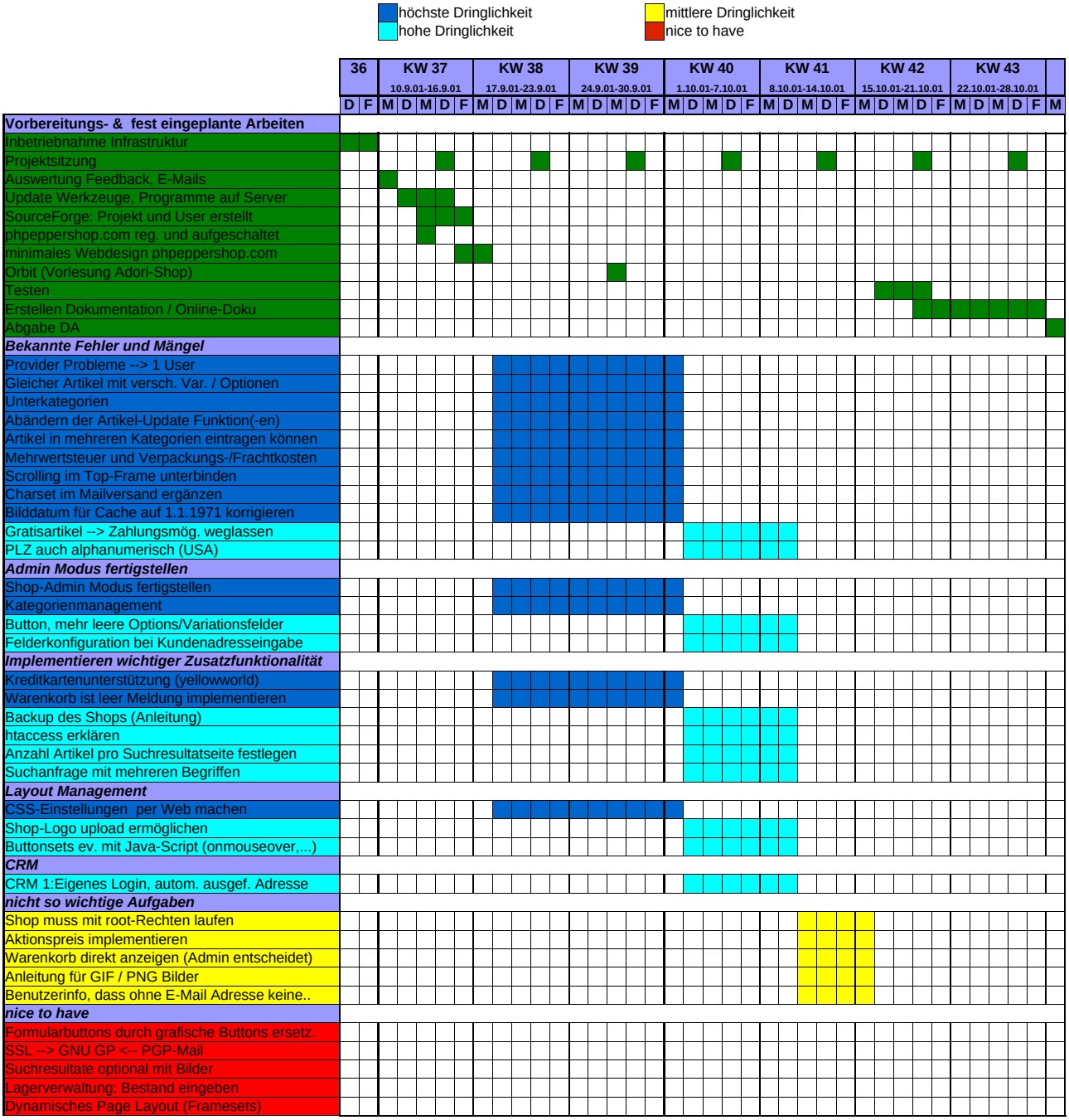
1	3	CSS-Einstellungen per Web machen
2	2	Shop-Logo upload ermöglichen --> Als (GIF-)Bilddatei im Bilder-Verzeichnis ablegen
3	2	Eigene Buttons (Buttonsets) ev. mit Java-Script (onMouseOver, onClick,...)
4	3	Dynamisches Page Layout mit einer Auswahl Framesets (auch: Header und Footer dynamisch)

CRM

1	3	CRM 1: Eigenes Login, automatisch ausgefüllte Adresse, Login zwingend / nicht zwingend --> Händlermodus

 © 2001 R. Glanzmann & J. Fontanil

1.6.2 Zeitplanung, erstellt zu Projektbeginn



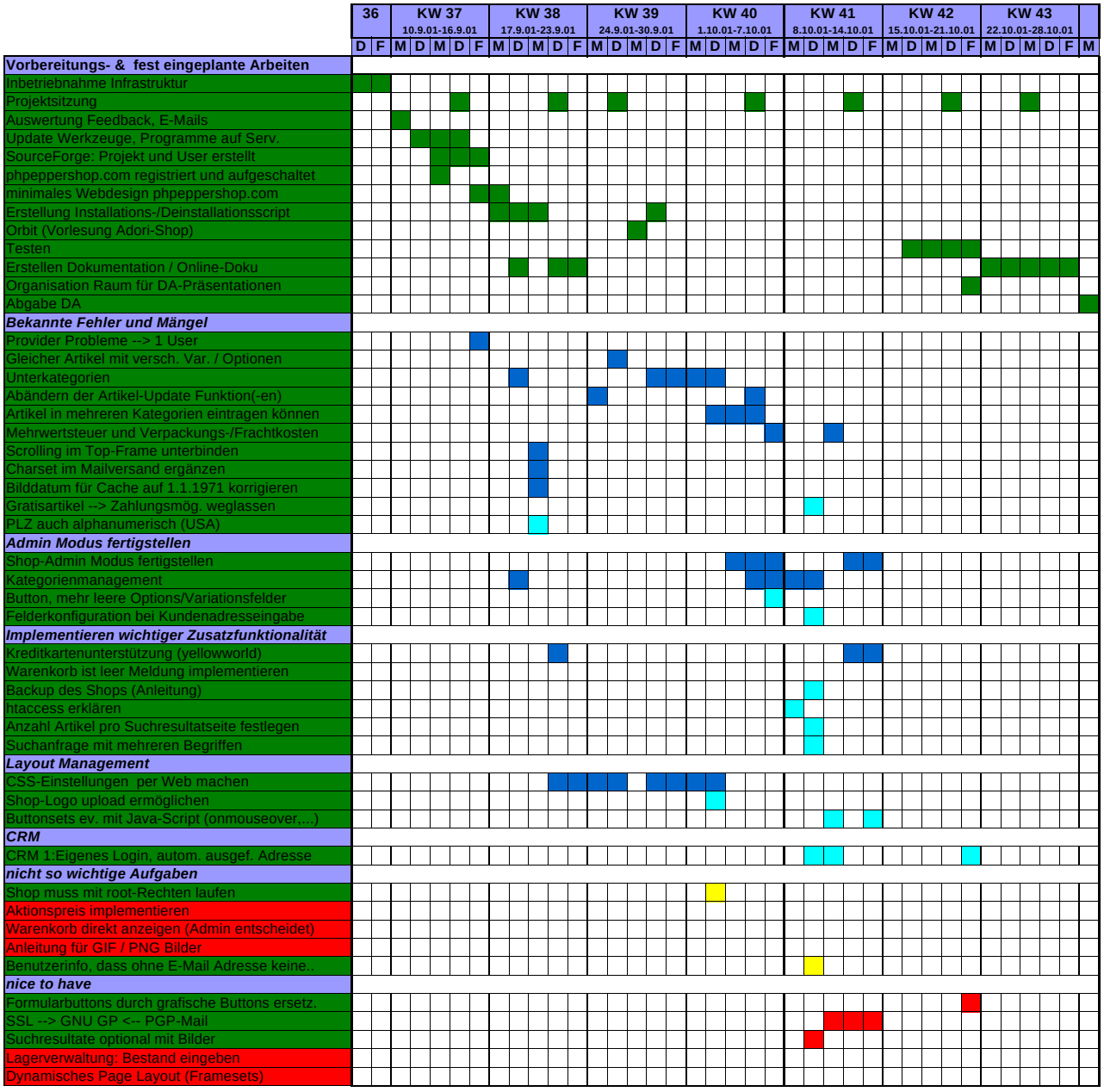
1.6.3 Effektiver Ablauf des Projekts

Wie in der folgenden Tabelle ersichtlich ist, haben wir fast alle geplanten Funktionen implementieren können. Die jetzt noch rot unterlegten Arbeiten stellten sich während der Implementierung des Shop für dessen Funktionalität und Bedienbarkeit als nicht elementar heraus, so dass wir während der Diplomarbeit deren Priorität heruntersetzten.

Einige niederprioritäre Funktionen konnten praktisch im gleichen Arbeitsschritt mit einer Funktion, der wir eine höhere Priorität zugeteilt hatten, gelöst werden. Dies erklärt, warum wir nicht konsequent eine Prioritätsstufe nach der anderen abarbeiteten.

 © 2001 R. Glanzmann & J. Fontanil

Arbeiten die erfolgreich gelöst werden konnten
Arbeiten, zu dessen Lösung keine Zeit mehr blieb oder die uns als unwichtig erschienen
Die Farben in der Wochenübersicht wurden aus der Zeitplanung vom 18.09.2001 übernommen und entsprechen den Prioritäten der Arbeiten



 © 2001 R. Glanzmann & J. Fontanil

1.7 Tests

Schon als wir mit Programmieren begannen, merkten wir, dass ein kompletter Test aller Funktionen und Zustände schlussendlich nicht mit einem vertretbaren Zeitaufwand möglich sein würde. Wir beschlossen einige Massnahmen, damit der Shop in seiner Schlussversion möglichst wenige, dazu leicht lokalisierbare Fehler enthielte.

1. Funktionen sollen, falls ein Problem auftritt, dieses direkt im Klartext detailliert beschrieben an den Browser ausgeben.
2. Funktionen, die den Rückgabewert eigentlich nicht brauchen, geben bei Erfolg „true“ zurück. Tritt irgend ein Problem auf, wird ein „false“ zurückgegeben. Der Fehler kann also recht genau lokalisiert werden.
3. Funktionen sollen direkt nach der Erstellung mit einigen sinnvollen Extremwerten getestet werden.
4. Schnittstellen zwischen Funktionen müssen genau definiert werden.
5. Jede Zeile, deren Bedeutung auf den ersten Blick nicht gleich erkennbar ist, soll kommentiert werden.

Wir wendeten dieses Konzept konsequent an. Es entstanden jedoch einige Funktionen, die aufgrund ihrer Grösse und Komplexität nicht mehr komplett ausgetestet werden konnten. Durch die oben beschriebenen Massnahmen war (und ist) es jedoch möglich, einen Fehler sehr genau zu lokalisieren.

1.7.1 Usability-Test

Da wir nicht von allen Anwendern das technische und logische Verständnis erwarten können, wie wir es mittlerweile haben, entschlossen wir uns, einen umfangreichen Usability-Test durchzuführen.

Da wir natürlich nicht selbst die Usability unseres Shops testen konnten, machten wir uns auf die Suche nach Testpersonen, die sich bereit erklärten, den Shop auf Bedienbarkeit hin zu prüfen und fanden diese dann schliesslich auch.

Andrea Harder, 9500 Wil:

- Testen des kompletten Administrationsbereichs auf Bedienbarkeit, Korrektheit und Verständlichkeit.
- Einkaufen als normaler Shopuser

Jolanda Jud, 9500 Wil:

- Betreiben eines bereits installierten und konfigurierten Shops (Artikel und Kategorienmanagement)
- Erstellen eines eigenen Shoplayouts
- Einkaufen als normaler Shopuser

Roland Brühwiler, 8560 Märstetten:

- Installation des Shops
- Verständlichkeit der technischen Einstellmöglichkeiten
- Einkaufen als normaler Shopuser

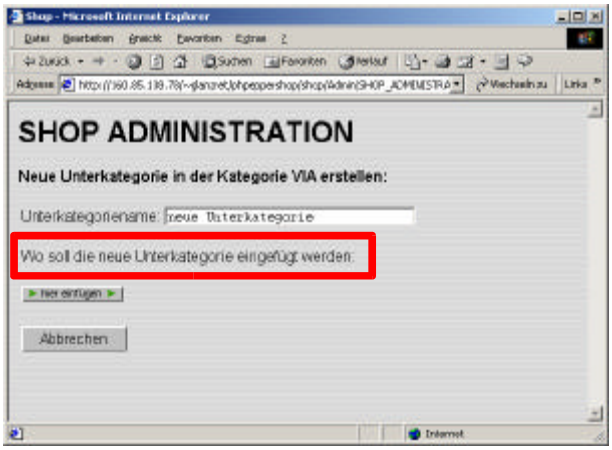
Was uns erfreute war, dass von unseren Testpersonen keine grösseren Fehler gefunden wur-

 © 2001 R. Glanzmann & J. Fontanil

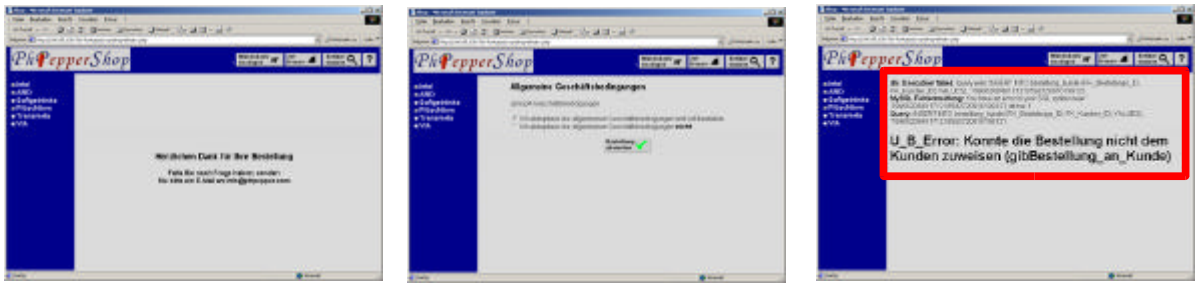
den. Einige kleine Fehler und Probleme konnten von uns schnell lokalisiert und behoben werden.

Betreffend Bedienbarkeit bemängelten alle Testpersonen die gleichen drei Punkte:

1. Wird eine Unterkategorie in einer Kategorie erstellt, in der noch keine Unterkategorien enthalten sind, so wird dem Administrator mitgeteilt, er könne wählen, wo die Unterkategorie eingefügt werden soll, obwohl eigentlich gar keine Auswahlmöglichkeit besteht:

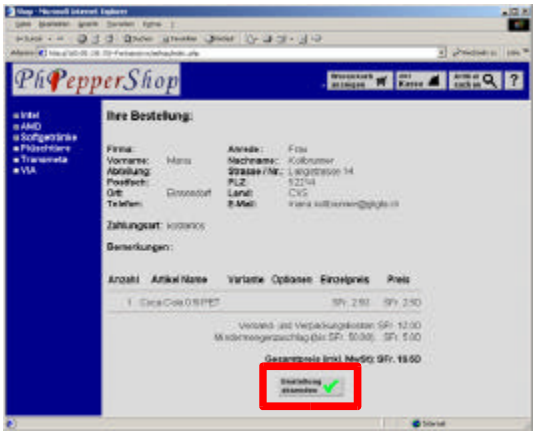

SHOP ADMINISTRATION
Neue Unterkategorie in der Kategorie VIA erstellen:
Unterkategoriename: neue Unterkategorie
Wo soll die neue Unterkategorie eingefügt werden:
Abbrechen

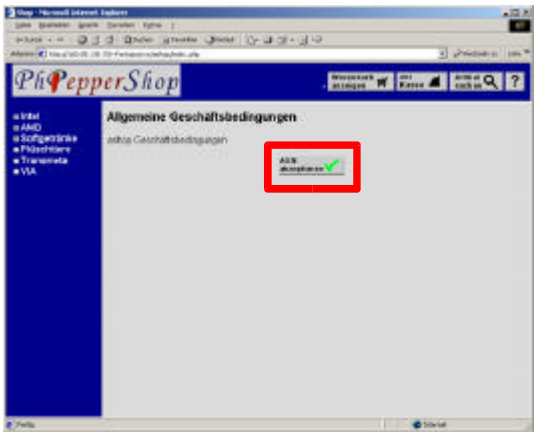
2. Drückt man nach dem Absenden der Bestellung den Zurück-Button des Browsers, und dann nochmals auf „Bestellung absenden“, wird erneut ein Bestätigungs-E-Mail versendet und eine Fehlermeldung wird an den Browser ausgegeben.



3. Es ist irritierend, dass nach dem Drücken des „Bestellung absenden“-Knopfs noch die Allgemeinen Geschäftsbedingungen angezeigt und akzeptiert werden müssen. Wenn man den Knopf „Bestellung absenden“ drückt, sollte man davon ausgehen können, dass die Bestellung unmittelbar danach auch abgesendet wird.

 © 2001 R. Glanzmann & J. Fontanil





Leider hatten wir nicht mehr die Zeit, alle drei Mängel zu beheben. Da wesentlich mehr Leute mit dem Shop selbst, als mit dem Administrationsbereich in Berührung kommen werden, legten wir dir Priorität ganz klar auf den Shop selbst.

Wir behoben die beiden letzteren Mängel und entschlossen uns, als Hinweis in die Bedienungsanleitung für den Administrator zu schreiben, dass das Kategorienmanagement vielleicht etwas verwirrend sein kann, wenn man eine Unterkategorie in eine Kategorie einfügt, die keine weiteren Unterkategorien enthält.

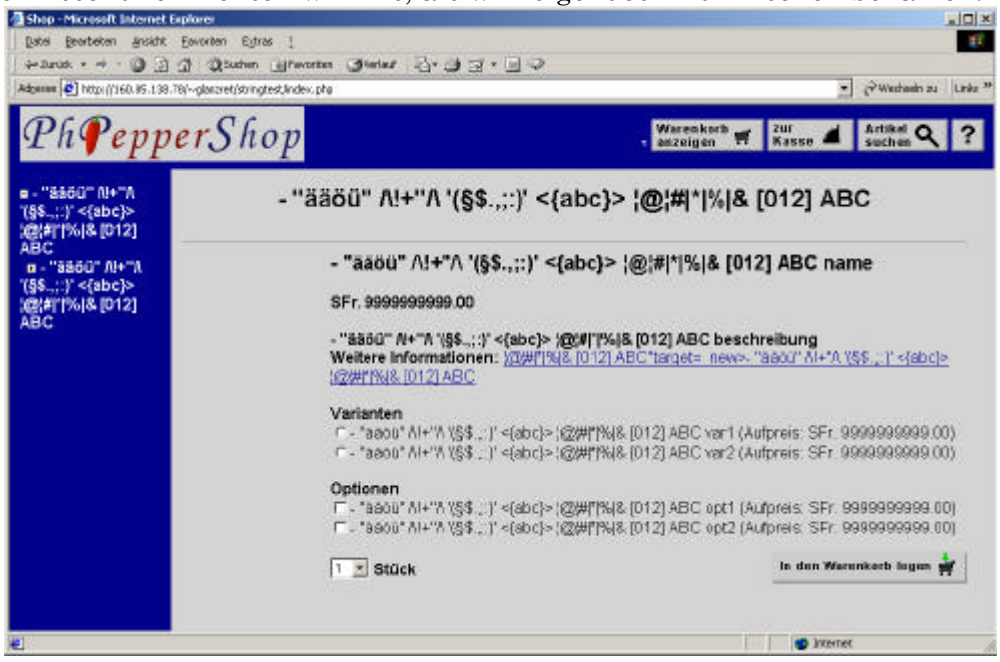
1.7.2 Extremwerte Test

Wir stellten bei vielen anderen Webshops fest, dass sich komische Effekte ergaben, wenn man in Formularfeldern gewisse Sonderzeichen einfügte. Dies wollten wir unbedingt verhindern.

Wir erstellten einen ziemlich gemeinen Teststring, der so ziemlich alle erdenklichen Sonderzeichen und Sonderzeichen-Kombinationen enthält:

```
- "ääöü" /\!+"/\ '(§$.,;:)' <{abc}> ¦@¦#|*|%|& [012] ABC
```

Wir verwendeten darauf ziemlich viel Zeit, um den Shop komplett Sonderzeichen-fähig zu machen. Schlussendlich freuten wir uns, als wir folgendes Bild zu sehen bekamen:



 © 2001 R. Glanzmann & J. Fontanil

Dies war der Beweis, dass alle Strings korrekt erfasst, verarbeitet und angezeigt wurden. Wir verwendeten den gleichen Test-String dann auch noch zum Sonderzeichentest des CRM- und Bestellungs-Teils. Auch diese Tests fielen positiv aus.

1.7.3 Funktionstest

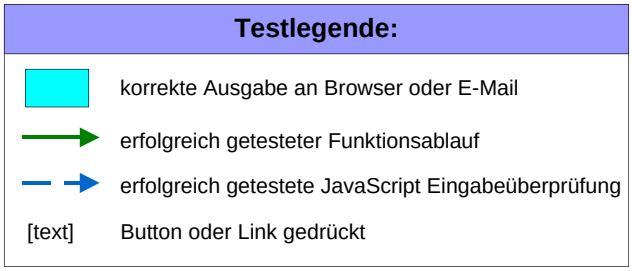
Bei den Funktionstests konzentrierten wir uns hauptsächlich auf die neu hinzugekommenen Funktionen des Shops; das Kundenmanagement, die Eingabeüberprüfung per JavaScript, Unterkategorien, Zahlungsablauf und das Layout-Management. Da die komplette Auflistung aller Tests den Rahmen dieses Dokuments sprengen würde, werden hier exemplarisch die wichtigsten Testfälle des CRM's und des Zahlungsablaufs, abgedruckt. Da es sich in den letzten beiden Projektarbeiten bewährt hatte, die Testfälle zur besseren Übersichtlichkeit zu visualisieren, verfolgten wir dieses Konzept auch hier weiter.

Der Shoptest wurde mit dem Internet Explorer 5.5 durchgeführt (Windows 2000). Vor Aufruf des Shops leerten wir den Browsercache.

Für die Richtigkeit der Testergebnisse,

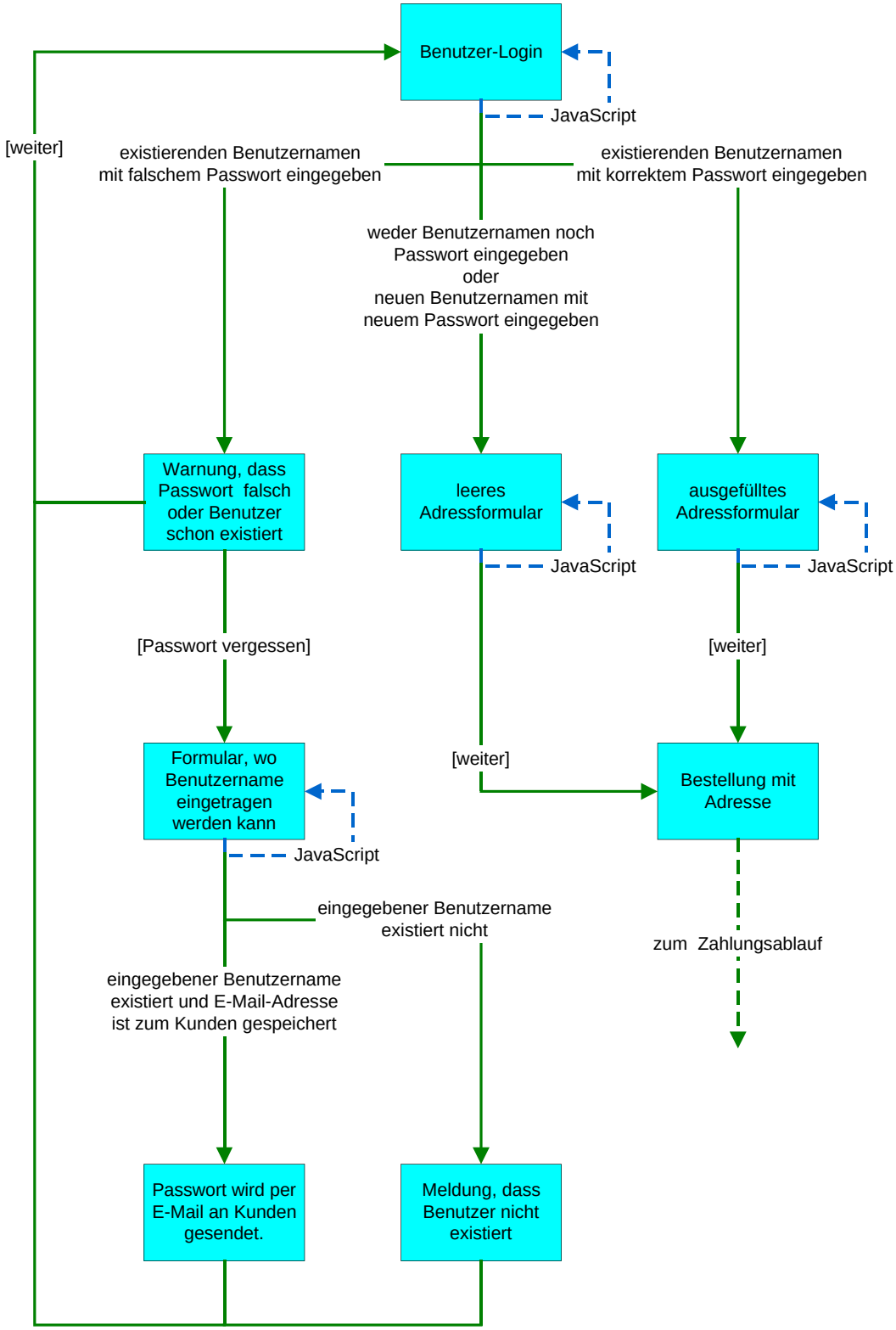
R. Glanzmann

Winterthur 26.10.2001



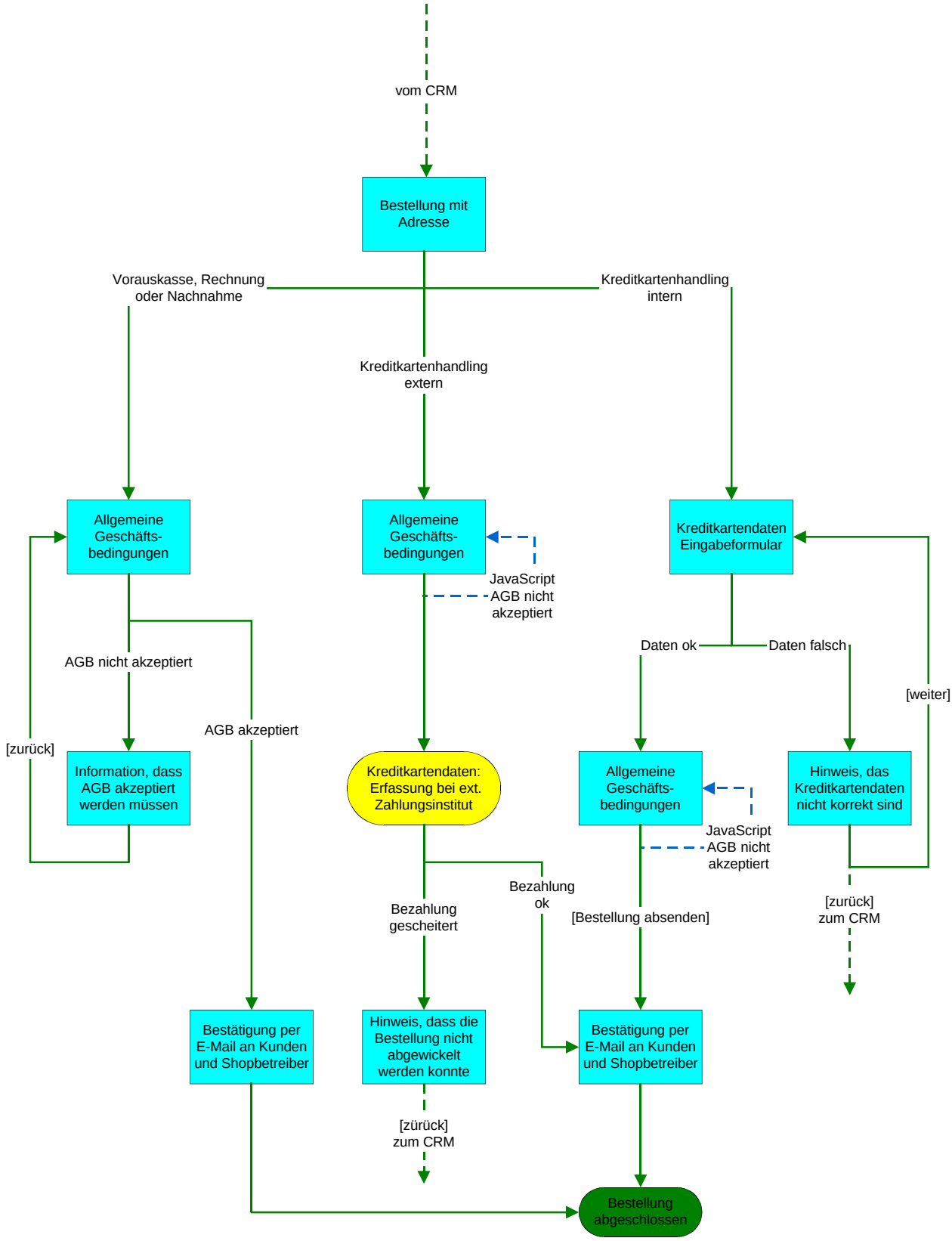
 © 2001 R. Glanzmann & J. Fontanil

Test des CRM's (Customer Relationship Management)



 © 2001 R. Glanzmann & J. Fontanil

Test des Bestellungsablaufs



 © 2001 R. Glanzmann & J. Fontanil

1.8 Erfahrungen

1.8.1 Installation der Systemumgebung - LAMPS

Inhalt dieses Kapitels

Da wir mit der aktuellen Version von PHP (v.4.0.6) arbeiten wollten, haben wir uns entschieden, den Apache Server, PHP und SSL-Unterstützung neu zu installieren. Damit unsere Erfahrungen, die wir hier machten, auch an andere Leute weitergegeben werden können, haben wir diesen Erfahrungsbericht im Stil einer umfassenden Anleitung geschrieben.

Wieso diese Kombination?

Alle fünf Teile eines LAMPS-Systems: Linux, Apache, MySQL, PHP, (open-, mod_)SSL sind Projekte, welche gemeinsam haben, dass sie Open Source sind. Sie kosten nichts, sind erprobt und sehr leistungsfähig. Alle Projekte werden aktiv weiterentwickelt. Mit Linux als kostenlosem Betriebssystem, erhält man heute über eine Distribution (SuSE, Redhat,...) ein zu kommerziellen Serverbetriebssystemen konkurrenzfähiges Produkt.

Apache ist ohnehin der am weitesten verbreitete Webserver der Welt. Er lässt hinsichtlich Stabilität und Erweiterbarkeit keinerlei Wünsche offen. Insbesondere die Kombination von Apache mit PHP ist heutzutage sehr populär und auch schon recht häufig anzutreffen. ...Ausserdem ist er nicht anfällig auf Code Red und Nimbda...

MySQL ist eine kleine, aber im Web-Umfeld weit verbreitete Datenbank. Sie bietet zwar keine referentielle Integrität und auch sonst kann sie nicht unbedingt mit einer IBM DB2 oder einer Oracle 9i konkurrieren. Ihr grosser Vorteil ist aber die grosse Unterstützung in der Web Open Source Gemeinde. Es gibt unzählige Projekte, welche MySQL als Datenbank nutzen. Dies liegt nicht zuletzt auch daran, dass MySQL sehr einfach zu handhaben ist. Hier sei in diesem Zusammenhang noch das Projekt phpMyAdmin (http://www.phpwizard.net/phpMyAdmin/) erwähnt, welches auf PHP basierend eine sehr komfortable Oberfläche zur Datenbankadministration bietet.

PHP ist eine leistungsfähige Skriptsprache, primär dazu ausgelegt im Web-Umfeld verwendet zu werden. PHP geniesst in seiner vierten Inkarnation eine breite Unterstützung, einerseits durch Webserver-Einbindungen, andererseits aber auch durch zahlreiche Funktionen, die inzwischen in PHP implementiert wurden. PHP ist einfach zu erlernen, benutzt die vielfach verwendete ANSI-C Syntax und unterstützt von Haus aus viele Datenbanken. Die für das Webscripting unumgänglichen String-Operationen hat es zum grössten Teil von Perl übernommen. Die Einbindung des PHP Codes in HTML erfolgt analog zu Microsofts ASP. Auf diese Weise wird von allen bekannten Sprachen und Techniken 'das Beste' verwendet und in PHP vereint.

Wenn man mit Geld hantieren will, kommt es auf Sicherheit an, und die Anforderungen an sie sollte man nicht unterschätzen. Die Datenübertragung von Kreditkartennummern, und Benutzerdaten und ihren Einkäufen dürfen für Dritte nicht einsehbar sein. Das heisst, dass man primär die Datenübertragung über das 'unsichere' Internet schützen, sprich verschlüsseln, muss. Eine heute anerkannte Technologie um dies zu realisiern ist SSL (Secure Socket Layer). OpenSSL ist eine Open Source Implementation der Verschlüsselungsfunktionalität, welche vom mod_SSL (Apache DSO-Modul) benutzt wird.

 © 2001 R. Glanzmann & J. Fontanil

Installation

Quellen beschaffen

Die beschriebenen Quellen beziehen sich auf Pakete, welche im September 2001 aktuell waren. Natürlich werden die Pakete zu einem späteren Zeitpunkt andere Versionsnummern tragen.

Linux

Linux beschafft man sich am besten als Komplettpaket in Form einer Distribution. Wir arbeiteten mit *SuSE Linux Professional 7.1*. Für welche man sich entscheidet, ist grundsätzlich Geschmackssache und im speziellen für diese Anwendung eigentlich nicht von Bedeutung. Wir benutzen hier weder das X-System, noch distributionsspezifische Eigenheiten. Vorausgesetzt wird lediglich ein lauffähiges Linux-System (natürlich mit gcc, flex, yacc, u.s.w.).

Wir empfehlen die folgenden Pakete gleich ins Verzeichnis /usr/local/src/ zu speichern, dann können wir vom gleichen Pfad ausgehen. Ausserdem ist so durch die Verzeichnisstruktur festgelegt, dass es sich um Sourcepakete handelt.

Apache

http://httpd.apache.org/dist/httpd/

MySQL

http://www.mysql.com/Downloads/MySQL-3.23/mysql-3.23.42.tar.gz

Bei MySQL entstehen ungefähr alle zwei Wochen neue Versionen. Deshalb hier noch die allgemeine Download Adresse (Achtung: Source-Files herunterladen, nicht Binaries):

http://www.mysql.com/downloads/

PHP

http://www.php.net/downloads.php

SSL

openSSL: http://www.openssl.org/source/

mod_SSL: http://www.modssl.org/source/

Erweiterungen

GD-Library: http://www.boutell.com/gd/

Zlib-Library: http://www.gzip.org/zlib/

IJG JPEG Software: http://www.ijg.org/

PDFlib: http://www.pdflib.com/pdflib/download/index.html

Freetype: http://sourceforge.net/project/showfiles.php?group_id=3157

Eigentliche Installation

Entpacken der Pakete

Die Pakete werden wie folgt entpackt:

 © 2001 R. Glanzmann & J. Fontanil

```
tar xvfz paketename.tar.gz
```

Nun gibt es jeweils Verzeichnisse, in denen sich die entpackten Tarballs (.tar.gz Dateien) befinden.

MySQL

Wir installieren zuerst MySQL, da die Datenbank unabhängig von den anderen Paketen ist.

Das Verzeichnis, in welchem MySQL seine Datenbanken (`data`) ablegt, kann mit dem Configure- Parameter `localstatedir` angegeben werden. Wir haben uns hier für den Pfad `/usr/local/mysql/data` entschieden. Mit dem Parameter prefix geben wir das Installationsverzeichnis unserer MySQL Datenbank an. (z.B. `/usr/local/mysql/3.23.42/` für unsere aktuelle MySQL Version). Nach der Installation erstellen wir noch einen Softlink. Auf diese Weise findet man nach der Installation einer neueren Version, die aktuelle unter `current`.

Im MySQL Source Verzeichnis:

```
./configure --prefix=/usr/local/mysql/3.23.42/ --localstate-
dir=/usr/local/mysql/data
make
make install
ln -s /usr/local/mysql/3.23.42 /usr/local/mysql/current
mkdir /usr/local/mysql
mkdir /usr/local/mysql/data
/usr/local/mysql/current/bin/safe_mysqld -u mysqladmin &
```

SSL Installation

Damit wir mit SSL arbeiten können, müssen wir unser System zuerst mit der SSL Funktionalität ausrüsten. Deshalb installieren wir zuerst openSSL:

Im openSSL Source Verzeichnis:

```
./config--prefix=/usr/local/openssl/0.9.6b
make
make test
make install
ln -s /usr/local/openssl/0.9.6b /usr/local/openssl/current
```

Apache und mod_SSL

Wir erstellen nun eine lauffähige Version von Apache, welche mit Dynamic Shared Objects umgehen kann. Dies ist eine überaus praktische Technologie. DSOs werden auch Apache Module genannt und funktionieren in etwa analog zu den bekannten Linux Modulen. Man kann sie einbinden und muss nicht für jedes neue Modul den ganzen Apache neu übersetzen.

Wir patchen die Quellen des Apache Webservers mit dem mod_SSL Paket. Gleichzeitig wird die Apache Configure Funktion aufgerufen. Das `datadir` Verzeichnis beherbergt das Webserver Hypertext Document Root. Wir haben uns hier für `/usr/local/apache/htdocs` entschieden.

Im mod_SSL Source Verzeichnis (*nicht* im Apache Source Verzeichnis!):

 © 2001 R. Glanzmann & J. Fontanil

```
./configure --with-apache=../apache_source_verzeichnis \
--with-ssl=../openssl_source_verzeichnis \
--prefix=/usr/local/apache/1.3.20 \
--datadir=/usr/local/apache/htdocs \
--enable-module-most \
--enable-shared-max \
--enable-module-ssl
```

Im Apache Source Verzeichnis:

```
make
```

Server-Zertifikat erstellen

Wir erzeugen nun ein Zertifikat zur späteren Verwendung von SSL-Sitzungen. Dieses wird von einer Certificate Authority (CA) unterschrieben. Wir legen die CA hier auch gleich mit an.

CA erstellen:

Im Apache Verzeichnis:

```
make certificate TYPE=custom
```

Signature Algorithm:	R
Country Name:	CH
State or Providence:	ZH
Locality Name:	Winterthur
Organisation Name:	PhPepperShop
Organisational Unit Name:	CA
Common Name:	PhPepperShop CA
Email Address:	ca@mydomain.com
Certificate Validity:	365
Certificate Version:	3

Nachdem nun der CA-Schlüssel erzeugt wurde, geht es ans Server-Zertifikat. Dabei ist zu beachten, dass der *Common Name* den Rechnernamen darstellt, unter welchem der Server später per SSL erreicht werden kann! (z.B. https://myshop.mydomain.com)

Country Name:	CH
State or Providence:	ZH
Locality Name:	Winterthur
Organisation Name:	PhPepperShop
Organisational Unit Name:	Webmaster
Common Name:	myshop.mydomain.com
Email Address:	info@mydomain.com
Certificate Validity:	365
Certificate Version:	3

Die Schlüssel sollen natürlich *verschlüsselt* werden, dies also bestätigen (zuerst CA, dann Server). Angemerkt sei hier noch, dass bei jedem Neustart des Apache das Passwort für den Server-Key eingegeben werden muss.

 © 2001 R. Glanzmann & J. Fontanil

Apache Installation vervollständigen

Im Apache Source Verzeichnis:

```
make install
ln -s /usr/local/apache/1.3.20 /usr/local/apache/current
```

Damit der Apache auch mit SSL-Unterstützung startet, müssen wir noch einen kleinen Eintrag in der Datei: /usr/local/apache/current/bin/apachectl ändern:

```
start)
if [ $RUNNING -eq 1 ]; then
    echo "$0 $ARG: httpd (pid $PID) already running"
    continue
fi
#Original:
#if $HTTPD ; then
#gaendert, -DSSL eingefuegt:
if $HTTPD -DSSL; then
    echo "$0 $ARG: httpd started"
else
    echo "$0 $ARG: httpd could not be started"
    ERROR=3
fi
;;
```

Nun müssen noch alle SSL Einträge aus der httpd.conf.default in die Datei httpd.conf kopiert werden. Es geht v.a. darum, die Module `libssl` und `modssl` zu laden (LoadModule).

PHP

PHP ist ein komplexes System und damit es wie von uns erwartet funktioniert, müssen wir zuerst noch ein paar Hilfslibraries kompilieren und installieren:

Freetype

Im Freetype Source Verzeichnis:

```
./configure --prefix=/usr/local
make
make install
```

PDFlib

Im PDFlib Source Verzeichnis:

```
./configure --prefix=/usr/local
make
make install
```

Zlib

Im Zlib Source Verzeichnis:

 © 2001 R. Glanzmann & J. Fontanil

```
./configure
make
make install
```

IJG JPEG Software

Im IJG JPEG Software Source Verzeichnis:

```
./configure
--enable-shared \
--enable-static \
--prefix=/usr/local
make
make test
make install
```

GD

Bevor wir die GD Library kompilieren können, müssen wir noch das Makefile editieren und abändern. Die Defaulteinstellungen benutzen nicht die libjpg, libpng, freetype Libraries. Das wollen wir aber und müssen deshalb die vorgegebenen Zeilen kommentieren und die Alternativen auskommentieren (CFLAGS und LIBS).

Bei den INCLUDEDIRS und den LIBDIRS sollte man jeweils einen Zusatz in der Form: `-I/usr/local/include/freetype2` beziehungsweise `-L/usr/local/lib` machen.

Nun sollte noch überprüft werden, ob auch folgende Pfade vorkommen:
```
INSTALL_LIB=/usr/local/lib
INSTALL_INCLUDE=/usr/local/include
INSTALL_BIN=/usr/local/bin
```

Und nun im GD Source Verzeichnis:

```
make
make install
```

PHP konfigurieren und kompilieren

Wichtig ist das `with-config-file-path`-Attribut. Im angegebenen Pfad wird PHP später nämlich seine `php.ini` suchen.

Im PHP Source Verzeichnis:

 © 2001 R. Glanzmann & J. Fontanil

```
./configure \
--with-apxs=/usr/local/apache/current/bin/apxs \
--with-mysql=/usr/local/mysql/current \
--with-zlib \
--with-ftp \
--with-gd \
--with-jpeg-dir=/usr/local/lib \
--with-pdflib \
--enable-versioning \
--enable-track-vars=yes \
--enable-url-inclueds \
--enable-sysvshm=yes \
--enable-sysvsem=yes \
--with-config-file-path=/etc
make
make install
```

Nachdem nun PHP installiert wurde, müssen wir nun noch eine Datei ins etc-Verzeichnis kopieren:

```
cp ./php.ini-dist /etc/php.ini
```

Es empfiehlt sich, nun in der `php.ini` den *SafeMode* einzuschalten. Dies macht man, indem man einen Editor (z.B. `pico`) nimmt und die Zeile sucht, in welcher `SafeMode=Off` steht und diesen dann auf `SafeMode=On` stellt.

Konfiguration

Wir müssen im Verzeichnis /usr/local/apache/current/conf/ die zentrale Konfigurationsdatei des Apache (httpd.conf) editieren. Zuerst sollten alle aktuellen Versionsnummern die im Skript vorkommen (bei uns z.B. 1.3.20) durch current ersetzt werden. Nun müssen wir noch die Mime-Types für PHP hinzufügen, beziehungsweise auskommentieren. Dies macht man am besten indem man folgende Zeile sucht, auskommentiert und .phtml noch hinzufügt:

```
AddType application/x-httpd-php     .php      .phtml
```

Zusätzlich konfigurieren wir Apache nun so, dass er auch PHP-Dateien als Default-Index Dateien akzeptiert. Dazu suchen wir den Eintrag DirectoryIndex und fügen noch die PHP Datei-Endungen hinzu:

```
DirectoryIndex    index.html    index.php    index.htm
```

Starten

Wir können nun den Apache Webserver starten, indem wir folgende Anweisung eingeben:

```
/usr/local/apache/current/bin/apachectl start
```

Wenn SSL korrekt installiert wurde, wird man an dieser Stelle angehalten seine Passphrase einzugeben. Danach startet der Apache Webserver.

Mit einem Webbrowser (zur Not auch per `lynx`), kann man nun schauen, ob der Webserver auch läuft.

 © 2001 R. Glanzmann & J. Fontanil

1.8.2 Einrichten eines Passwort geschützten Bereichs mit htaccess

Diese Anleitung soll lediglich das Prinzip, wie man ein Verzeichnis mittels htaccess schützt, erklären. Sie kann nicht als detaillierte Installationsanleitung verwendet werden, da sich die notwendigen Schritte je nach eingesetztem Webserver und Hostingprovider unterscheiden. Alle Pfadangaben beziehen sich explizit auf unseren Testserver und müssen durch eigene ersetzt werden.

Das Prinzip des Verzeichnisschutzes per htaccess besteht aus vier Schritten:

1. Im Webserver-Konfigurationsfile (typisch: `httpd.conf`) muss eingestellt werden, dass die Konfigurationsanweisungen in `.htaccess`-Dateien beachtet werden sollen. Dies geschieht mittels des `AllowOverride` Eintrags auf das gewünschte Verzeichnis des Benutzers:

   ```
   <Directory "/home/fontajos/htdocs/">
   AllowOverride All
   </Directory>
   ```

   `Directory` gibt das Verzeichnis an, im welchem die Konfigurationsdatei beachtet werden soll.

2. In dem Verzeichnis, welches geschützt werden soll, muss die Datei `.htaccess` mit folgendem Inhalt erstellt werden:

   ```
   AuthUserFile /home/fontajos/.htpasswd
   AuthName "Administrationsbereich"
   AuthType Basic
   <Limit GET POST>
   require valid-user
   </Limit>
   ```

   `AuthUserFile` definiert, wo sich das Passwortfile befindet, welches die Benutzernamen und Passwörter der für den Zugriff berechtigten Benutzer enthält.
   `AuthName` enthält den Text, der im Login-Fenster ausgegeben wird.
   Mit **`AuthType`** setzt man den gewünschten Authentifikationstyp fest, wobei entweder Basic oder Digest angegeben werden kann. Prinzipiell sollte man die Digest-Variante bevorzugen, da hier, im Gegensatz zur Basic-Authentifikation, das Passwort niemals im Klartext über die Verbindung geschickt wird und somit von einem potentiellen Angreifer nicht in Erfahrung gebracht werden kann. Obwohl die Digest-Authentifikation sicherer ist, wird sie bisher noch nicht von allen Browsern unterstützt!

3. Mit dem Programm `htpasswd` kann eine Passwortdatei erstellt werden. Um eine Passwortdatei anzulegen, muss zuerst ermittelt werden, wo sich `htpasswd` befindet (`which htpasswd` oder `locate htpasswd`). Danach kann mit folgendem Befehlsaufruf eine Passwortdatei, die einen Benutzer enthält, erstellt werden:

   ```
   verzeichnis_von_htpasswd/htpasswd -c .htpasswd Benutzername
   ```

   `htpasswd` ist der Dateiname, wie die Passwortdatei heissen soll.
   Als **`Benutzername`** kann ein beliebiger Name eingegeben werden.

   Mit folgendem Aufruf kann der Passwortdatei noch ein weiterer Benutzer zugefügt, oder das Passwort eines schon enthaltenen Benutzers geändert werden:

 © 2001 R. Glanzmann & J. Fontanil

```
verzeichnis_von_htpasswd/htpasswd .htpasswd Benutzername
```

4. Schlussendlich muss der Webserver noch dazu bewogen werden, die Konfigurationsfiles neu einzulesen. Entweder man „killt“ die Prozesse des Webservers (`killall -HUP httpd`) oder startet den Webserver neu (`./apachectl restart`).

 © 2001 R. Glanzmann & J. Fontanil

1.8.3 SourceForge einrichten

Einleitung

In den folgenden Kapiteln beschreiben wir chronologisch unsere Erfahrungen, wie wir unser Projekt auf SourceForge eingerichtet haben, welche Hürden wir nehmen mussten und auf was man achten muss, wenn man selber ein Projekt auf SourceForge einrichten will. Wir wollen diesen Erfahrungsbericht als in sich geschlossenes Dokument veröffentlichen, sodass auch andere Leute davon profitieren können.

SourceForge bietet Open Source Projekten eine kostenlose, sehr umfangreiche und international ausgerichtete Entwicklungsumgebung an. Geboten wird unter anderem ein CVS Repository, Mailinglisten, eine Homepage, Newsgroups und vieles mehr.

SourceForge einrichten

Eigener Account

Um ein Open Source Projekt überhaupt erst lancieren zu können, muss man sich zuerst unter http://ww.sourceforge.net/ anmelden. Dies macht man am einfachsten mit einem Mausklick links oben auf 'Neuer Benutzer (SSL)'. Die folgenden einzugebenden Felder sind selbsterklärend. Wenn man dann sein Login hat und sich einloggt, findet man auf der linken oberen Ecke eine Option 'Neues Projekt'. Auf diese Weise kann man sein neues Projekt einrichten.

Ein neues Projekt lancieren

Nun muss man dem SourceForge Terms of Service Agreement zustimmen. Vor allem Punkt 6 und 7 sind interessant. Unter anderem wird dort beschrieben, dass SourceForge keinerlei Haftung für Datenverlust übernimmt. Es ist also trotz allem ratsam immer ein aktuelles Backup zuhause zu haben. Der Punkt 7 der TOS widmet sich der Account- respektive Project Termination. Hierbei sei nur erwähnt, dass man grundsätzlich nicht von SourceForge verbannt wird, wenn man sich halbwegs ordnungsgemäss verhält. Wir wollen hier nicht alle Paragraphen besprechen, aber Punkt 5 finden wir erwähnenswert: Man stimmt mit dem Akzeptieren dieses Vertrags zu, dass man mit den Ressourcen von SourceForge keinerlei kommerzielle Ziele verfolgt.

Nun muss man seinem Projekt zuerst einmal einen Namen geben und es auf Englisch beschreiben. Diese Beschreibung ist Grundlage für die Zulassung des Projekts durch ein Beurteilungsteam von SourceForge. Wenn es unter einer Open Source Lizenz steht und frei verfügbar ist, sollte das aber ohne Probleme klappen. Weiter muss noch eine Lizenz gewählt werden (GPL oder LGPL dürften wohl die am meisten verwendeten Lizenzen sein) und es muss eine kleine Beschreibung angegeben werden. Die Public Description kann nur 255 Zeichen lang sein und wird bei Suchresultaten zu Projekten mit angezeigt. Zu guter letzt muss man noch einen Unix-Namen des Projekts angeben (die beschriebenen Restriktionen beachten).

Jetzt braucht man ein bisschen Geduld. Unser Projekt wurde innerhalb von 12 Stunden akzeptiert und von SourceForge eingerichtet, das kann aber zuweilen auch bis zu 24 Std. dauern. Wenn man sich nun wieder einloggt, findet man unten rechts sein Projekt.

Das neue Projekt einrichten

Nach einem Klick auf das Projekt befindet man sich auf der Projekt Übersichtsseite. Von hier

 © 2001 R. Glanzmann & J. Fontanil

aus komplettiert man zuerst einmal die Projekt-Konfiguration. Man hab eine reiche Auswahl an angebotenen Dienstleistungen, welche alle konfiguriert und eingerichtet werden wollen.

Übersicht des Projekts

Summary
Überblick | Admin | Homepage | Forums | Tracker | Bugs | Support | Patches | Mailinglisten | Aufgaben | Dokumentation | Neuigkeiten | CVS | Filesgroup
short_forum Foren |

Anfangen sollte man beim 'Admin'-Teil. Im Misc. 'Project Information'-Fenster sieht man in welchem Verzeichnis sich was befindet. Bei uns sah die Verzeichnisstruktur wie folgt aus:

Homepage Link:	phpeppershop.sourceforge.net
Group shell (*SSH*) server:	phpeppershop.sourceforge.net
Group directory on shell server:	/home/groups/p/ph/phpeppershop
Project WWW directory:	/home/groups/p/ph/phpeppershop/htdocs

Man hat also sicher schon mal eine Homepage für das Projekt (100MB Webspace, CGI und PHP-Scripts sind auch erlaubt), ein CVS Repository (dazu später mehr) und Zugriff auf einen Mailinglisten-Server. Natürlich darf auch eine Shell nicht fehlen. Mit Telnet kommt man aber nicht weit, SourceForge erlaubt klugerweise nur SSH (SSH1 und SSH2).

Wenn man sich das erste mal auf die Shell des SourceForge Servers einloggt, (`ssh projektname.sourceforge.net`) wird einem der Fingerprint des jeweiligen Servers angezeigt. Man sollte sich kurz die Zeit nehmen und diesen mit dem jeweiligen Fingerprint hier vergleichen. Das beste Sicherheitstool nützt nichts, wenn man es nicht richtig anwendet:

```
       Host            Protocol                Key Fingerprint
shell.sourceforge.net    SSH1    33:50:33:91:b4:ec:7a:b6:fa:39:99:b0:ff:65:63:a7
shell.sourceforge.net    SSH2    4c:68:03:d4:5c:58:a6:1d:9d:17:13:24:14:48:ba:99
cvs.sourceforge.net      SSH1    ff:f1:42:ca:ac:ae:a7:c8:e8:52:45:65:a1:a0:0a:b6
cvs.sourceforge.net      SSH2    02:ab:7c:aa:49:ed:0b:a8:50:13:10:c2:3e:92:0f:42
```

Bevor man weiterfährt, sollte man sich gleich einmal einloggen und zwar auf dem 'Projekt Server': `projektname.sourceforge.net` und ganz wichtig auch auf dem CVS-Server `cvs.projektname.sourceforge.net`. Man wird das CVS-System *nicht* benutzen können, wenn man sich nicht vorher einmal per SSH darauf eingeloggt hat (erst da wird das benötigte Home-Verzeichnis erstellt).

Trove Categorisation

Man kann sich jetzt der Trove Categorisation seines Projekts widmen (auf [Edit] klicken). Hier kann man sein Projekt beschreiben und kategorisieren, sodass Leute, die nach Projekten mit gewissen Eigenschaften suchen, auch fündig werden. Wenn ein Projekt nicht eindeutig zugeordnet werden kann, darf man mehrere Einteilungen benutzen. SourceForge erlaubt bis zu drei Einteilungen gleichzeitig.

Public Information

Im Admin-Bereich klicken wir nun auf 'Edit Public Info'.

Admin | User Permissions | Edit Public Info | Project History | VHOSTs |
Edit/Release Files | Post Jobs | Edit Jobs | Edit Multimedia Data | Database Admin

Hier sieht man die öffentlich zugänglichen Dienste für dieses Projekt und die Beschreibung dazu. Hier muss natürlich ein jeder selber wählen, was für sein Projekt alles benötigt wird. Mailinglisten, Foren und CVS wird wohl jeder wählen, Surveys wahrscheinlich eher weniger.

 © 2001 R. Glanzmann & J. Fontanil

Es ist erstaunlich, was SourceForge einem hier alles anbietet, und zwar gratis und franko. Bleibt nur die Frage, ob das von VA Linux gesponserte System auch für immer bestehen bleibt (216 Mio. $ Verlust bei 16 Mio. $ Umsatz im letzten Quartal...) Unserer Meinung nach wäre SourceForge eine Einrichtung, die vom Staat getragen werden müsste.

Entwickler einrichten

Als nächstes werden alle am Projekt beteiligten Mitarbeiter eingefügt und ihren Usern die entsprechenden Rechte geben:

1. Im Group Members Fenster den entsprechenden Usernamen eingeben und 'Add User' wählen.
2. In den 'User Permissions' (oben) kann man dem jeweiligen User Rechte und Ämter geben.

Webseiten für das Projekt aufsetzen

Jedes Projekt braucht natürlich seine Homepage. Hier wollen wir nicht darauf eingehen wie man seine Page gestalten sollte, sondern nur, wohin man sie ablegen muss und einige Punkte, welche man nicht vergessen sollte.

In der Projektübersicht (im Admin Modus) sieht man den Project WWW Directory Path. Dorthin muss man seine Seiten ablegen.

Am besten loggt man sich in SourceForge per SSH ein:

```
ssh -l Username Projektname.sourceforge.net
```

Danach wechselt man in das Project WWW Directory. Dort kann man dann mit dem Secure Copy Befehl von SSH die Daten hochladen. Beispiel:

```
scp Datei Username@shell.sourceforge.net:/home/groups/Projektname/htdocs
```

```
scp test.pl fontajos@shell.sourceforge.net:/home/groups/p/ph/phpeppershop/htdocs
```

Das sollte auf einer Projekt-Webpage alles enthalten sein:

- Einen Link auf die SourceForge Projektseite (z.B. http://sourceforge.net/projects/phpepper-shop/)
- Links zu den Mailinglisten (die gleich angelegt werden)
- SourceForge Icon
- Einen Link zum Webinterface des CVS-Servers: http://cvs.sourceforge.net/cgi-bin/cvsweb.cgi/?cvsroot=Projektname)
- Projektspezifische Daten (z.B. ein Howto für CVS oder ähnlich)

Wer direkt Projektdaten auf der Homepage darstellen will, kann dies auch tun. Genaueres findet man hier: https://sourceforge.net/docman/display_doc.php?docid=1502&group_id=1

Mailinglisten anlegen

Mailinglisten und Foren sind die Kommunikationsmittel der verteilten Entwickler. Es wird empfohlen mindestens folgende drei Standard-Listen für sein Projekt anzulegen:

- announce Hier werden neue Versionen angekündigt
- devel Eine Mailingliste für Entwickler
- users Dasselbe für die Benutzer

Mailinglisten anlegen und verwalten kann man über das 'Mailinglisten' Menü in der Projektübersicht. Wenn man auf 'Admin' klickt, kann man neue Mailinglisten anlegen oder be-

 © 2001 R. Glanzmann & J. Fontanil

stehende administrieren (ist sie öffentlich, Beschreibung des Zwecks, ...).

Man hat noch die Wahl, seine Mailinglisten im 'GNU Mailman' zu administrieren. Dort kann man sich mit dem per E-Mail erhaltenen Passwort einloggen und seine Mailingliste ganz detailliert und komfortabel konfigurieren.

News-Foren

News-Foren zum Projekt lassen sich per Klick auf 'Forums' in der Projektübersicht administrieren. In den Admin Teil kommt man über 'Admin'. Dort kann man die Foren konfigurieren. Drei News-Foren waren für uns schon automatisch eingerichtet worden.

- Developers
- Help
- Open Discussion

Es wurde noch ein weiteres, spezielles Forum angelegt: **Neuigkeiten**. Der Zweck dieses Forums ist analog zur announce-Mailingliste.

Dokumentation

Wie koordiniert man die Dokumentation der einzelnen Funktionen und Module, welche von verschiedenen Entwicklern von irgendwo auf der Welt programmiert wurden? SourceForge bietet hier mit der Dokumentationsfunktionalität eine gute Lösung. Jeder Entwickler kann seine Dokumentation in verschiedenen Gruppen organisiert (z.B. Deutsch, Englisch,...) posten und jeder hat danach Einsicht, so muss nicht alles im Code sein.

Eine kleine aber wichtige Anmerkung dazu: Die Dateien die upgeloaded werden, müssen zuerst von einem Administrator begutachtet und für gut befunden werden, erst dann sind sie für alle ersichtlich. Auch muss man aufpassen, wenn man Plain-Text Berichte uploaden will. Alle Umbrüche sind dann weg, weil das Dokument wie eine HTML-Seite betrachtet wird (Stichwort
-Tag). Auf diese Weise hat man aber gute Gestaltungsmöglichkeiten.

(Bug-, Support-, Patch-, Feature-Request-) Tracker

Der Tracker ist ein Support-Management Tool. Man kann mit ihm 'Tickets' erstellen und an die Entwickler des Projekts senden. Für den betroffenen Entwickler, erscheint dieses mit einer Priorität versehen und nach Themen sortiert auf seiner personalisierten Einstiegsseite.

Um den Tracker zu konfigurieren kann man als Projekt-Administrator von der Projekt Übersichtsseite aus auf 'Tracker' klicken und dort die entsprechende Sparte wählen. Wenn man jetzt auf 'Admin' klickt, kann man den Tracker konfigurieren. Es ist wichtig, dass man das am Anfang macht, da die Leute ihre Meldungen sonst nicht sinnvoll einordnen können. Ausserdem geht unkonfiguriert jedes Ticket an alle Admins, was meistens wohl auch nicht erwünscht ist.

ACHTUNG: Einmal eingerichtete Kategorien und Gruppen können *nicht* mehr entfernt werden.

Die einzelnen Tracker-Listen können über 'update preferences' noch zusätzlich konfiguriert werden Unter anderem kann man dort auch festlegen, ob diese Liste überhaupt zur Verfügung stehen soll.

So, jetzt kommt nur noch CVS.

 © 2001 R. Glanzmann & J. Fontanil

CVS einrichten

Was ist CVS?

Wenn man mit CVS (Current Versions System: http://www.cvshome.org/) noch keine Erfahrung hat, braucht man ein bisschen Eingewöhnungszeit. Diese sollte man aber tunlichst auf sich nehmen. Die Vorteile einer CVS-gestützten Software-Entwicklung gegenüber einer ohne CVS, sind vergleichbar mit einem Zirkus-Seiltänzer ... mit und ohne Netz! Wer einmal mit CVS gearbeitet hat, wird es garantiert nicht mehr missen wollen.

Was macht CVS eigentlich? CVS führt Buch über die Veränderungen, die am Sourcecode vorgenommen wurden, zusammen mit einer Beschreibung dessen, was verändert wurde. Man kann sich zu jeder Zeit von jedem Ort die aktuelle und jede vorangegangene Version jeder Datei im Repository ansehen. Das CVS-Repository ist eine Datenbank, in welcher alle Daten abgelegt werden. Durch CVS können mehrerer Entwickler zur gleichen Zeit an der gleichen Datei arbeiten und ihre Änderungen zusammenführen. Wenn zwei Änderungen sich gegenseitig beeinflussen, werden diese Entwickler darauf hingewiesen und es wird sichergestellt, dass der Konflikt behoben wird, bevor die neue Version der Datei im Repository aufgenommen wird.

CVS Entwicklungszyklus

Im Wesentlichen besteht der Entwicklungszyklus aus folgenden drei Schritten:

1. Man importiert den initialen Sourcecode ins Repository. Anschliessend checkt jeder Entwickler eine Arbeitskopie aus dem Entwicklungsbaum aus.
2. Nun arbeitet jeder Entwickler in seiner gewohnten Umgebung am Programm. Sobald ein neues Feature fertig ist, aktualisiert man zunächst die lokale Kopie, um auf dem laufenden zu sein und schreibt die veränderten Dateien zurück in den Entwicklungsbaum. Siehe dazu auch 'Update-Policy' weiter unten.
3. Sollten beim Zurückschreiben der Veränderungen irgendwelche Konflikte auftreten, wird man darauf hingewiesen und muss anschliessend manuell entsprechende Änderungen vornehmen. In den Dateien werden die veränderten Bereiche mit ">>>>>" hervorgehoben und beide Versionen angezeigt. Hier löscht man entweder einfach die alte Version, oder man passt den Bereich an, damit er wie gewünscht arbeitet. Wenn alle Konflikte beseitigt sind, geht man wieder zurück zu Punkt 2 und setzt seine Arbeit fort.

CVS für sein Projekt auf SourceForge einrichten

Um mit CVS zu arbeiten und es für sein Projekt auf SourceForge einzurichten, benutzt man verschiedene Kommandozeilen Befehle. Wenn es jemand vorzieht unter Windows mit CVS zu arbeiten, so kann er z.B. WinCVS verwenden. WinCVS bietet noch nicht die komplette Funktionalität des UNIX Kommandozeilen Pendants, es kann aber für den täglichen Gebrauch gut verwendet werden. Weitere Angaben zu WinCVS und SourceForge findet man unter: http://sourceforge.net/docman/display_doc.php?docid=766&group_id=1. Für die weitere Beschreibung gehen wir auf die UNIX Kommandozeilen Implementation von CVS ein.

Bevor man nun seinen bestehenden Code ins CVS Repository von SourceForge importiert, noch ein wichtiger Hinweis: Die Verzeichnisstruktur, welche man dem CVS System übergibt, ist 'für die Ewigkeit' gedacht. Man kann nicht einfach wieder ein Verzeichnis löschen, nachdem man es importiert hat! Wir mussten z.B. den Support von SourceForge darum bitten unser CVS-Repository entsprechend zu ändern, nachdem uns ein Faux-pas passiert ist. Man sollte sich deshalb vorher genau überlegen, was man auf CVS veröffentlicht und wie man das

 © 2001 R. Glanzmann & J. Fontanil

gerne strukturiert haben möchte.

Weiter macht es Sinn, nicht einfach den Source Code in das CVS-Root-Verzeichnis des jeweiligen Projekts abzulegen. Man sollte vielmehr eine kleine Struktur verwenden. Die meisten Projekte haben mehrere Verzeichnisse und manchmal sogar noch Informations-Dateien im CVS-Root. Sinnvoll erscheint uns ein Verzeichnis namens `/src` für den Source Code und ein weiteres für die Dokumentation, meistens `/doc` genannt. Hier lohnt es sich, einmal bei schon bestehenden Projekten vorbeizuschauen und sich anzusehen, wie sich diese Projekte organisiert haben. Wenn man Dateien hinzufügt, welche lediglich informativen Charakter besitzen, sollte man diese mit Grossbuchstaben schreiben, da sie von UNIX dann am Anfang der Dateiauflistung dargestellt werden.

SourceForge akzeptiert, wie schon weiter oben erwähnt, nur SSH Übertragungen. Folglich muss auch bei CVS SSH verwendet werden. Um CVS mitzuteilen, dass es SSH benutzen soll, ist es, eine Variable zu exportieren (man kann auch `setenv` (Linux: `export`) verwenden):

```
export CVS_RSH=ssh
```

Damit dieser Eintrag nach jedem Login zur Verfügung steht, empfiehlt es sich, diese Zeile der Datei `/etc/profile.local` hinzuzufügen.

Den Quellcode ins Repository importieren (Import)

Einen CVS Import macht man grundsätzlich nur zu Beginn eines Projekts, um den vielfach schon vorhandenen Code ins CVS-Repository aufzunehmen.

Zuerst wechselt man in das Verzeichnis, dessen Unterverzeichnisse und Dateien ins CVS-Repository importiert werden sollen. Alles was jetzt im aktuellen Verzeichnis und in allen Unterverzeichnissen vorhanden ist, wird ins Repository übernommen!

Man muss folgendes Kommando benutzen um den Code zu importieren:

```
cvs -d:ext:Loginname@cvs.Projektname.sourceforge.net:/cvsroot/Projektname
import Verzeichnisname vendor start
```

Wobei der Loginname und der Projektname entsprechend ersetzt werden müssen. Mit dem Verzeichnisnamen ist das Verzeichnis gemeint, in welchem die Daten auf dem CVS-Repository abgelegt werden sollen.

Nachdem man das Passwort eingegeben hat, erhält man noch die Möglichkeit, eine Message einzugeben (es wird der UNIX Editor `vi` geöffnet). Wenn man nichts mitgeben möchte, tippt man `:q!` Um den Editor wieder zu verlassen. Wer eine Message eingeben möchte, sollte sich doch zuerst mit dem vi-Editor befassen.

Arbeitskopie aus dem Repository auschecken (Checkout)

Ein grosser Vorteil von CVS ist, dass man von überall auf der Welt an seinem Projekt weiter arbeiten kann. Man muss lediglich ein Checkout machen und schon hat man seine aktuelle Arbeitskopie des Projekts.

Als eingetragener Entwickler:

```
cvs -d:ext:Loginname@cvs.Projektname.sourceforge.net:/cvsroot/Projektname
co Verzeichnisname
```

Wobei hier auch wieder der Loginname und der Projektname entsprechend ersetzt werden müssen. Mit dem Verzeichnisnamen ist das Verzeichnis gemeint, welches man vom Repository herunterladen möchte.

 © 2001 R. Glanzmann & J. Fontanil

Als anonymer Benutzer

```
cvs -d:pserver:anonymous@cvs.Projektname.sourceforge.net:/cvsroot/Projektname
login
```

Nachdem man sich nun als anonymous eingeloggt hat, kann man das eigentliche Checkout machen:

```
cvs   -d:pserver:anonymous@cvs.Projektname.sourceforge.net:/cvsroot/Projekt-
name
co Verzeichnisname
```

Hier muss auch der Loginname und der Projektname entsprechend ersetzt werden. Mit dem Verzeichnisnamen ist das Verzeichnis gemeint, in welchem die Daten auf dem CVS-Repository abgelegt werden sollen.

Upgedatete Datei zurück ins Repository schreiben (Update, Commit)

Hier geht es darum ein weiter entwickeltes File, oder mehrere Dateien, zurück in das CVS-Repository zu schreiben.

Man sollte sich vorher aber auf eine Update-Policy einigen. Diese Policy sollte regeln, *wann* es sinnvoll ist, eine neue Version ins Repository zurück zu schreiben. Wir haben uns z.B. darauf geeinigt, dass wir immer erst dann ein Update machen, wenn wir eine neue lauffähige Version unseres Moduls oder unserer Funktion haben.

Da man mit dem Checkout-Befehl schon alle Angaben zum Projekt angegeben hat (-d) muss man sie jetzt nicht mehr explizit angeben:

```
cvs update
cvs commit -m "Message"
```

An die Message kann man eine kleine Nachricht mitgeben. Hier drin beschreibt man in kurzen Worten, was man am Code geändert hat. Es gibt auch Projekte, welche hier eine Message-Policy eingeführt haben.

Eine neue Datei ins Repository einfügen (Add, Commit)

Um eine neue Datei ins CVS-Repository aufzunehmen, muss man zuerst einen Checkout des Repositories lokal vorhanden haben. Darin plaziert man zuerst lokal die neue Datei am richtigen Ort im Verzeichnisbaum, programmiert vielleicht schon die Funktionalität hinein und gibt dann folgende Kommandos ein:

```
cvs add [-kb] Dateiname
cvs commit -m "Message"
```

Mit der Message kann man wiederum analog zum Update eine beschreibende Nachricht mitgeben. Die Option `-kb` gibt CVS an, dass es sich um eine binäre Datei handelt. Man muss binäre Dateien (z.B. Word-Dateien, ...) als solche kennzeichnen, weil CVS nicht-binäre Dateien liest und gegebenenfalls gewisse Tags ersetzt.

Webzugriff auf das Repository (Diff, ...)

Um Unterschiede zwischen Versionen von Dateien begutachten zu können, haben wir uns entschieden, das von SourceForge bereitgestellte ViewCVS einzusetzen. Dieser Service erlaubt es, neben weiteren Funktionen, das CVS Repository über das Web anzusehen. Über http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/Projektname/ oder auch über die Projekt-Übersichtspage, kommt man darauf. Hier kann man bequem zwischen den Versionen der einzel-

 © 2001 R. Glanzmann & J. Fontanil

nen Dateien hin und her wechseln, sie miteinander vergleichen, ihre Anzeige nach bestimmten Kriterien sortieren und man hat einen sehr schönen Überblick über das Projekt-eigene CVS-Repository. Es gibt natürlich für all diese Funktionen auch entsprechende Kommandozeilen-Befehle (siehe Anhang).

Eine Datei oder ein Verzeichnis entfernen

Datei: Dies ist ein heikler Punkt von CVS. Schliesslich will man, auch wenn man eine Datei löscht, nicht deren Entwicklungsbaum verlieren. CVS bietet deshalb einen remove-Befehl an, der die Datei zwar faktisch als gelöscht markiert, sie aber nicht physikalisch vom Repository entfernt. Folgende Schritte muss man abarbeiten um eine Datei zu 'löschen':

1. Sicher stellen, dass es keine noch nicht commiteten Änderungen für das File gibt.
2. Die Datei lokal löschen (rm Dateiname)
3. cvs remove Dateiname
4. cvs commit Dateiname

Verzeichnis: Eigentlich gibt es keinen Weg für SourceForge Benutzer, Verzeichnisse zu löschen, es gibt aber auch hier einen Weg: Man löscht zuerst wie oben beschrieben alle Dateien im Verzeichnis (dabei können auch Wildcards wie z.B. * benutzt werden). Nun kann man, wenn man entweder einen Checkout oder ein Update macht, die Option -P mit angeben, was für CVS heisst, dass es leere Verzeichnisse von der Arbeitskopie entfernen soll. Es wird empfohlen *immer* mit der Option -P zu arbeiten.

Eine Datei verschieben

Um eine Datei zu verschieben, muss man lediglich die alte Datei löschen (siehe oben) und die Neue am neuen Ort erstellen. Hier ein Beispiel:

```
mv old new
cvs remove old
cvs add new
cvs commit -m "old zu new umbenannt" old new
```

CVS Id-Tag

Wenn man ein Programm mit Hilfe von CVS verwaltet, sollte man im Header jeder Programm-Datei folgenden Eintrag als Kommentar einfügen:

```
$Id$
```

CVS wird dann nach jedem Commit an dieser Stelle die aktuelle Versionsnummer, Filenamen, Datum, Zeit und Usernamen einfügen. Bei uns sieht das zum Beispiel so aus:

```
$Id: artikel_def.php,v 1.7 2001/10/26 06:57:48 fontajos Exp $
```

 © 2001 R. Glanzmann & J. Fontanil

Weitere Services von SourceForge

Aufgaben (Task Management)

Wir sind hier nicht genauer auf das Task-Management eingegangen, weil wir es für die Zeit während unserer Diplomarbeit nicht benutzt haben. Dieser Service ist aber ab einer gewissen Projektgrösse unverzichtbar. Mit ihm kann ein Projekt Administrator noch zu erledigende Aufgaben publizieren und den Entwicklern zuweisen. Auf diese Weise ist immer ersichtlich, wer an welchem Problem arbeitet und was noch zu tun ist.

Surveys

Man kann im Rahmen seines Projekts auch Umfragen starten (Edit Public Info).

Compile Farm

SourceForge bietet seinen Benutzern diesen wirklich aussergewöhnlichen Service an. SourceForge bietet seinen Benutzern Zugriff auf verschiedene Netzwerke und damit auf verschiedene Host Systeme mit unterschiedlichsten Betriebssystemen. Auf diese Weise kann man sein Produkt auch auf anderen Betriebssystemen und Architekturen testen. Weiter lassen sich so bequem spezielle Binaries des eigenen Projekts erstellen.

MySQL Datenbank

Jedem Projekt wird auf Anfrage eine MySQL Datenbank gegeben. Diese kann frei benutzt werden. Oft wird sie dazu benutzt, eine dynamische Webseite des Projekts zu unterstützen.

Virtual Hosts Service

Man kann SourceForge als Hoster benutzen. Per Webinterface kann man eine Anfrage an SourceForge senden, auf einen gewissen DNS Namen zu antworten.

Fragen zu SourceForge

Hier findet man viele Dokumente zu SourceForge: http://sourceforge.net/docman/?group_id=1

 © 2001 R. Glanzmann & J. Fontanil

1.8.4 Browser Kompatibilität

Alle Teile des Webshops wurden von uns mit den folgenden Browsern getestet:

Browser älterer Generation:

- Netscape 4.7x

Aktuelle Browser:

- Internet Explorer 5.x / 6
- Mozilla 0.9x
- Netscape 6.0 / 6.1

Bei der Wahl der zu Testzwecken verwendeten Browser beschränkten wir uns auf kostenlos erhältliche Produkte. Den Internet Explorer 5.x/6 sowie Netscape 6.x wählten wir, weil sie momentan die populärsten Browser darstellten. Netscape 4.7 brauchten wir, um eine gewisse Rückwärtskompatibilität zu wahren und Mozilla wird unserer Ansicht nach in Zukunft kontinuierlich an Beliebtheit gewinnen.

Positiv zu erwähnen ist, dass alle Browser neuster Generation die von uns erzeugten HTML-Seiten mit wenigen Ausnahmen korrekt interpretierten und darstellten. Die Definitionen in der Stylesheet-Datei wurden ordnungsgemäss ausgewertet. Der Mozilla-Browser besitzt jedoch eine wesentlich langsamere JavaScript-Engine, was uns beim Kategorienmanagement zwang, anstatt Mouse-Over, Mouse-Click-Ereignisse für die Aktualisierung des Menüs zu verwenden. Die JavaScript Aktivität reduzierte sich dadurch wesentlich, wodurch auch mit Mozilla eine akzeptable Geschwindigkeit erreicht wurde.

Ein weiterer Unterschied herrscht zwischen den Browsern Mozilla und Internet Explorer bei der Übergabe von Funktionsargumenten an eine JavaScript Funktion. Die JavaScript Engine vom Internet Explorer betrachtet die Werte, wie wenn sie mit Escape-Sequenzen codiert wären und entfernt diese selbständig. Da der Mozilla dies nicht macht, resultiert aus dem übergebenen Wert ein völlig anderes Resultat, falls im String ein oder mehrere Backslashes enthalten sind. Als Abhilfe für dieses Problem bleibt entweder die Möglichkeit, auf Sonderzeichen zu verzichten, oder die Funktionsargumente mit der PHP-Funktion `urlencode` zu encodieren (`urlencode` ersetzt alle Zeichen durch die Zeichenfolge %HH, wobei HH den Hexcode des betreffenden Zeichens im ASCII-Zeichensatz darstellt). So codierte Funktionsargumente müssen in der JavaScript Funktion dann natürlich wieder decodiert werden.

 © 2001 R. Glanzmann & J. Fontanil

Wir hätten uns viele Probleme ersparen können, wenn wir auf die Kompatibilität zum Netscape 4.7x Browser verzichtet hätten. Da jedoch ein nicht vernachlässigbarer Teil der Surfer noch Browser dieser Generation einsetzt, entschlossen wir uns, den Shop auch für diese Browsergeneration brauchbar zu machen. Die Probleme begannen damit, dass Netscape 4.7 eine Stylesheet-Datei, die sich nicht im gleichen Verzeichnis wie das Webdokument befand, schlichtweg nicht interpretierte. Diesem Problem entgegneten wir, indem wir die Stylesheet-Datei in jedes Verzeichnis kopierten, in dem sich Webdokumente befanden. Die nächste Schwachstelle von Netscape 4.7x ist, dass Stylesheets nicht ordnungsgemäss vererbt werden. Wenn wir also für eine Tabelle eine Schriftart definierten, mussten wir diese Stylesheet-Angabe für das Spalten- und Zeilen-Tag zusätzlich nochmals definieren (überladen). Ausserdem musste beim Verwenden von CSS-Klassen (z.B. `body.class, body.main, body.left`) im HTML-Dokument bei jedem Tag noch explizit die Klasse angegeben werden (müsste auch vom Body-Tag geerbt werden!). Das grösste Problem bestand jedoch bei der Interpretation der Stylesheet-Angaben für Links (<a>-Tag). Diese wertet der Browser überhaupt nicht aus. Die einzige Lösung, die wir hierfür fanden war, die Stylesheet-Angaben im HTML-Dokument mittels `style="..."` direkt in das Link-Tag zu schreiben.

Da CSS-Stylesheets ausserordentlich nützlich sind, um shopweit ein einheitliches Erscheinungsbild zu erhalten, wollten wir keinesfalls darauf verzichten. Wir nahmen uns deshalb die Mühe, die Stylesheet-Datei so zu erstellen, dass auch Netscape 4.7 damit zurecht kam. Durch mehrfaches Überladen von Styles und Klassen gelang uns dies dann nach langem experimentieren auch. Dies ist insofern sehr wichtig, da wir sonst das Layoutmanagement nur mit sehr viel Zeitaufwand hätten implementieren können.

1.8.5 PHP und Safe-Mode

PHP stellte sich wiederum (wie schon bei der Projektarbeit) als sehr nützliche Scriptsprache heraus, welche die meisten für die Web-Programmierung benötigten Funktionen schon mitbringt und so eine sehr effektive Software-Entwicklung ermöglicht.

PHP besitzt einen sogenannten Safe-Mode. Ist dieser eingeschaltet, läuft PHP in einem sichereren Modus. Viele Operationen auf das Filesystem und die Ausführung von Programmen durch PHP sind dann jedoch nicht mehr möglich. Um eine gewisse Sicherheit zu haben, schalten die meisten Provider diesen Modus ein. Da wir schon in der Projektarbeit konsequent mit eingeschaltetem Safe-Mode entwickelten, behielten wir dies auf für die Diplomarbeit bei.

Theoretisch ist es per Safe-Mode möglich, an allen Stellen auf das Filesystem zu schreiben, bei denen der Inhaber der Datei gleich dem Inhaber des PHP-Scripts ist. Dies erwies sich jedoch in der Praxis als falsch, worauf wir uns nach einer anderen Lösung umschauen mussten. Wir fanden dann heraus, dass man auf eine Datei, die schon existiert und volle Lese und Schreibrechte für jedermann besitzt, auch bei eingeschaltetem Safemode schreiben kann. Da sich die Anzahl der zum Shop gehörenden Dateien nicht verändert, erstellten wir Dummy-Files, für alles was man per Layoutmanagement und Artikelmanagement hochladen kann. Ein kleines Beispiel: Das Hintergrundbild des Hauptframes ist nach der Installation des Shops in der Datei `bg_main.gif` gespeichert. Trotzdem ist schon ein entsprechendes Pendent im JPG-Format vorhanden (`bg_main.jpg`). Lädt der Shopadministrator nun ein Hintergrundbild vom Typ JPG hoch, so wird das bereits existierende `bg_main.jpg` zum Überschreiben geöffnet. Auf diese Weise können trotz eingeschaltetem Safe-Mode Dateien hochgeladen werden.

Den zweiten Konflikt mit dem Safe-Mode hatten wir, als wir zur Verschlüsselung des Bestätigungs-E-Mails mittels Execute-Kommando das Programm GNU-PG aufrufen wollten. Der

 © 2001 R. Glanzmann & J. Fontanil

Safe-Mode verweigerte konsequent die Ausführung von Programmen per PHP. Wir fanden dann heraus, dass man in der PHP-Konfigurationsdatei (`php.ini`) ein Verzeichnis angeben kann (`safe_mode_exec_dir`), in dem sich die Programme befinden, die von PHP ausgeführt werden dürfen. Dies klappte dann mit anderen Programmen auch, GNU-PG konnte jedoch die Keyringe nicht richtig einlesen. Dieses Problem konnten wir leider bis jetzt nicht lösen. Da die Ausführung von GNU-PG per Shell funktioniert, wäre es vielleicht ein Lösungsansatz, den GNU-PG Aufruf in ein Shell-Script zu kopieren und dieses dann per PHP auszuführen. Wir werden uns sicher nach Abgabe der Diplomarbeit an Werner Koch, den Programmierer von GNU-PG, wenden und versuchen abzuklären, ob das Problem an GNU-PG oder beim Safe-Mode liegt.

Das letzte Problem, welches wir mit dem Safe-Mode hatten, war, dass ein Shell-Aufruf von PHP aus keine Pipe enthalten durfte. Aufrufe im Stil von `echo hallo welt | irgend_ein_programm` wurden nur bis zum Pipezeichen (|) interpretiert, was beim obigen Beispiel eine Ausgabe von „hallo welt“ an den Browser zur Folge hatte. PHP bietet jedoch auch für dieses Problem Abhilfe mit den Funktionen: popen (erstellt eine Pipe), pclose (schliesst eine Pipe), fputs (schreibt auf eine Pipe) und fgets (liest von einer Pipe).

Abschliessend kann zu PHP gesagt werden, dass es sicher die Richtige Scriptsprache für Webanwendungen ist. Wie allgemein bekannt ist, hat jedoch auch hier Sicherheit ihren Preis.

1.8.6 GNU-PG

Da für einen kleineren Webshop die Kreditkartenabrechnung per Mailorder-Vertrag unserer Meinung nach die lukrativste ist und diese Zahlungsart neu mit Visa- und Eurocard/Mastercard Karten abgewickelt werden kann, entschlossen wir uns, einen Versuch mit GNU-PG zur Verschlüsselung der Bestellungs-E-Mails mit dem PGP-Verfahren zu starten.

Wie bereits im Kapitel 1.8.5 zu lesen ist, gelang es uns jedoch bei eingeschaltetem Safe-Mode nicht, einen beliebigen Text mittels GNU-PG per PHP-Aufruf zu verschlüsseln.

Folgender Aufruf funktionierte in der Shell tadellos:

```
echo "dieser text wird verschlüsselt" | gpg -a --always-trust --homedir ".gnupg" --batch --no-secmem-warning -e --keyring .gnupg/pubring.gpg --secret-keyring .gnupg/secring.gpg -r "glanzret@zhwin.ch"
```

Parameter	*Erklärung*
-a	Ausgabe mit ASCII-Hülle versehen
--always-trust	Nimmt an, dass allen verwendeten Schlüsseln vertraut werden kann.
--homedir	Ort des Homeverzeichnisses
--batch	Stapelmodus: Keine Abfragen
--no-secmem-warning	Keine Memory-Sicherheits-Warnung ausgeben
-e	Daten verschlüsseln
--keyring	Als öffentlichen Schlüsselbund mitbenutzen..
--secret-keyring	Als geheimen Schlüsselbund mitbenutzen..
-r	Verschlüsseln für Name..

 © 2001 R. Glanzmann & J. Fontanil

 © 2001 R. Glanzmann & J. Fontanil

1.8.7 Strukturen beim Webprogramming

Wenn man ein Programm fürs Web entwickelt, muss man sich im klaren sein, dass gewisse starre Aufgabenteilungen und Unpässlichkeiten vorhanden sind. Es gibt viele Unterschiede zu Programmen, welche lokal auf dem Rechner laufen. In diesem Kapitel schildern wir unserer Erkenntnisse, welche wir in dieser Beziehung gewonnen haben.

Variablentransport zwischen Funktionen

Um Variablen zwischen Funktionen zu transportieren, arbeiteten wir von der Projektarbeit her viel mit Arrays und assoziativen Arrays. Auf diese Weise kann man bis zu zwei Variablen 'pro Zeile' transportieren. Das Problem entstand für uns immer dann, wenn von einem Objekt plötzlich mehrere Attribute verwendet wurden. Wir standen dann vor der Entscheidung, entweder die Funktionen neu anzupassen oder ein Objekt auf mehrere Array-Einträge zu verteilen. Schwieriger ist es, wenn man eine Funktion hat, welche plötzlich nicht nur ein Objekt, sondern gleich mehrere zurück geben sollte.

Ein Beispiel dafür wäre die getgesuchterArtikel($Suchstring,...)-Funktion (USER_ARTIKEL_HANDLING.php). Sie gab in der Shop-Version der Projektarbeit einen Array von Artikel-Objekten zurück. Nachdem ein Artikel nun in mehreren Kategorien sein kann, hätten wir die Klasse Artikel abändern müssen, was viele schon bestehende und gut funktionierende Funktionen betroffen hätte (getArtikel,...). Wir hätten auch den Array zur Übertragung benutzen können (ein Artikel, seine Kategorien, nächster Artikel,...). Wir haben uns aber dafür entschieden, eine neue Klasse 'Artikelmitkategorien' einzuführen. Wir schrieben also lediglich ein anderes Objekt in den Rückgabearray und konnten die Änderungen fast ausschliesslich in dieser Funktion lokal behalten.

Was hier breit an einem Beispiel erklärt wurde, ist genau genommen eine einfache Grundüberlegung: "***Erstelle Klassen für alles was du transportieren willst***".

Eine weitere elegante Möglichkeit besteht darin, mit Objekten zu arbeiten, welche andere Objekte enthalten, um auf diese Weise die logische Struktur von Daten abzubilden. Ein Beispiel hierfür wäre unser Bestellungsobjekt: Wir haben die get- und set-Funktionen so ausprogrammiert, dass die übergeordneten Objekte die untergeordneten beinhalten und somit die logische Struktur der Daten wiederspiegeln: Eine Bestellung enthält immer auch (mindestens) einen Artikel. Wir haben die getBestellung-Funktion so programmiert, dass man ein Bestellungsobjekt erhält, welches in seinem Artikelarray alle Artikel beinhaltet.

Unser Merksatz hier lautet: "***Bilde die Datenbankrelationen und logischen Assoziationen in den Objekten nach und erstelle die Transport-Funktionen dafür***".

Die ganze objektorientierte Entwicklung hat eben schon seine Vorteile. Rückblickend gesehen, hätten wir unseren Shop vollständig objektorientiert entwickeln sollen.

Wenn man am Designen von neuen Funktionen ist, welche mit Objekten interagieren und diese auch weitergeben, so sollte man sich bei der Überlegung, was ich der nächsten Funktion übergeben soll, immer folgende Fragen stellen:

Frage	*Design*
Sind die übertragenen Daten sicherheitskritisch?	Nur die ID des Objekts übergeben. Bsp. Artikel-Preise: Ein Shop-Kunde könnte den Preis nach seinem Gusto verändert übertragen.

 © 2001 R. Glanzmann & J. Fontanil

Frage	*Design*
Ist die Funktion / das Modul Performance kritisch?	- Wenn ein Datenbank-Zugriff vermieden werden kann: Nicht einfach ein ganzes Objekt üertragen, sondern nur die wirklich benötigten Attribute eines Objekts weiterreichen. - Wenn sowieso auf die Datenbank zugegriffen werden muss: SQL-Query optimieren (Joins anstatt mehreren SELECTs, SQL-Zugriffe minimieren, Datenmengen minimieren (nur die benötigten Attribute aus der Datenbank auslesen).
Wenn ich kann, sollte ich:	Mit nummerischen IDs arbeiten. Sobald Buchstaben, Benutzereingaben oder sogar Sonderzeichen übertragen werden müssen, wird die Übertragung mühsam und fehleranfällig (siehe Kapitel 1.8.8).

Andere Fehlermeldungen

Wenn man Webprogramming mit PHP, JavaScript und HTML betreibt, so muss man sich darüber im Klaren sein, dass man nicht den Komfort einer integrierten Entwicklungsumgebung wie z.B. Microsoft Visual C++ erwarten darf. Dies betrifft vor allem die Fehler-Rückmeldungen vom Programm zum Programmierer.

Wir hatten zu Beginn der Entwicklung zum Teil enorm Mühe, die Programmier-Fehler aufzuspüren, einfach weil wir nicht wussten, wo wir überhaupt Rückmeldungen erhalten. Wir haben ein paar dieser Fehlerfälle zusammengestellt und dokumentiert, wie wir dann jeweils vorgegangen sind.

Fehler	*Vorgehen*
PHP Code geändert, Reload, Fehler des Browsers: leere Seite.	Quellcode der Page ansehen, oft wird die HTML-Seite nicht angezeigt. Man sieht aber im Quellcode, wo das Programm ins Stocken geriet (SQL-Fehler, o.ä.).
HTML-Code geändert, Reload, Fehler des Browsers: leere Seite.	Quellcode der Page ansehen, überprüfen ob allfällige Tabellen, der Body und das HTML-Dokument selbst geschlossen wurden.
Neue Datei hinzugefügt, erster Zugriff, Fehler des Browsers: Zugriff nicht erlaubt.	Sind die Dateiattribute richtig gesetzt? Dürfen alle User auf die Datei zugreifen? (ideal: chmod 755).
Bild in der Datenbank geändert, Browser zeigt aber immer noch altes an	Meist ist der Cache des Browsers schuld. Vor allem beim Netscap 4.7x muss oft manuell der Cache gelöscht werden, bevor er das gleichnamige neue Bild richtig anzeigt.

 © 2001 R. Glanzmann & J. Fontanil

Fehler	*Vorgehen*
Formular gemacht, Submit Button funktioniert aber nicht (kein Ereignis nach Klick).	Ist das Formular komplett? <form action=...>....<input...> </form>. Wenn eines dieser Elemente fehlt, funktioniert das Formular nicht. Diese Situation kann vorkommen, wenn das Formular dynamisch von verschiedenen Funktionen zusammen gebaut wird.
Internet Explorer zeigt eine Seite mit Tabelle und / oder Formular an, Netscape 4.7x aber nicht.	Meistens ist die Tabelle oder das Formular nicht richtig geschlossen. Netscape 4.7x zeigt eine Tabelle erst an, wenn sie komplett eingelesen wurde.

Da es in den meisten Programmen keinen richtigen Debugger gibt, ist man auf echo "" und die Test Ausgabe der interessanten Variablen angewiesen.

Interaktivität einer HTML-Seite

Der erste grosse Unterschied zwischen lokaler Programmentwicklung und Webprogramming besteht im Funktions-Handling. Wenn ich lokal ein Programm entwickle, schreibe ich eine Eingabemaske und rufe danach mit den eingegebenen Variablen eine Funktion auf. Wenn man dieses Beispiel auf eine Webanwendung mit HTML und einer serverseitigen Scriptsprache wie z.B. PHP überträgt, merkt man, dass es ein gravierendes Problem gibt: Ich kann zwar eine Maske darstellen (HTML-Formular), aber ich kann danach nicht einfach eine PHP-Funktion aufrufen! Weiter geht es nur über einen HTML-Link. Entweder ist die URL im action= Attribut des Formular-Tags <form ..> definiert, oder sie ist in Form eines gewöhnlichen Links im Anchor-Tag vorhanden. Man merke sich: "***Von einer Webpage zur anderen geht's nur über eine URL weiter***". Diese Erkenntnis ist ziemlich wichtig!

Um dieser Problematik Herr zu werden, haben wir im PhPepperShop die sogenannte Aufruf-Schicht eingeführt (siehe Kapitel 2.2.1, Übersicht über die PHP-Funktionen und Module). Wir erstellten HTML-Seiten, welche PHP if- und elseif-Verzweigungen beinhalten. Nach jeder HTML-Page ruft man je nach Workflow sich selbst ($PHP_SELF) oder eine andere PHP-Datei auf und übergibt die darstellen Variable mit einer Nummer. Je nach Nummer kommt man so in einer Datei (bei uns Modul genannt) an verschiedene Orte, von wo aus dann PHP-Funktionen aufgerufen werden, welche dann auf den Daten operieren.

Ein weiterer Unterschied zwischen lokalen Programmen und Webanwendungen ist, dass man bei lokalen Programmen meistens ereignisgesteuerte Programm-Elemente innerhalb einer dargestellten View (Seite) hat. Hier muss man zuerst einmal ziemlich umdenken. In Webanwendungen arbeitet man primär mit der Seitenbeschreibungssprache HTML. HTML ist nicht dafür entwickelt worden, ereignisgesteuerte Operationen starten zu können. HTML Seiten sind in sich statisch. Wenn ich also z.B. ein Texteingabefeld für einen Artikelpreis habe, so kann ich nicht nach der Eingabe eines Textes prüfen, ob dieser Text auch alphanummerische Zeichen beinhaltet. Die Auswertung mittels PHP kann frühestens nach Aufruf einer weiteren HTML-Seite erfolgen und ist somit für gewisse Aufgaben zu starr. Hier setzt JavaScript an. Mit JavaScript kann man innerhalb einer HTML-Page dynamische, ereignisgesteuerte Elemente benutzen.

 © 2001 R. Glanzmann & J. Fontanil

Ein weiteres Problem ist die Verzweigung des Workflows, je nach Benutzereingabe auf der gleichen Seite. Hierzu mehrere Problemstellungen und wie man sie lösen kann:

GUI-Elemente	*Wie weiter*
Ich habe mehrere Knöpfe, der Shopbenutzer wird, je nachdem auf welchen Knopf er drückt, auf eine andere Seite geführt.	Jeden Knopf (IMG-Tag) in einen <a href=".."> Tag setzen. Somit hat man mit jedem Knopf einen eigenen Link erstellt, welcher gegebenenfalls Variablen per GET weiterreichen kann (2kB-Grenze bei GET nicht vergessen!).
Textfelder und / oder Radio-Buttons und / oder Checkboxen und / oder Pull-Down Menüs. Mit einem Knopf geht es weiter.	HTML-Formular erstellen. Man kann den Submit-Knopf auch als Grafik darstellen mit dem Tag: <input type=image src="..">

Entwurf von neuen Komponenten für den PhPepperShop: (Entwicklungszyklus)

Während der Diplomarbeit erweiterten wir das von der Praktischen Arbeite begonnene Framework für einen Webshop in PHP kontinuierlich zum PhPepperShop. Wir ergänzten den Shop komponentenweise um neue Funktionen. Dabei ist uns bewusst geworden, welche Elemente des PhPepperShops betroffen werden, wenn man ihm eine neue Funktionalität hinzufügen will. Wir haben heraus gefunden, dass folgende Überlegungen und Implementierungsschritte nacheinander abgearbeitet werden müssen, damit man eine neue Funktion oder sogar Komponente effizient und fehlerlos einbauen kann:

1. **Sich fragen**:
    1. Was brauche ich? (Use-Cases helfen hier!)
        - Datencontainer (Klassen)
        - Datenbankzugriffsfunktionen (get, new, upd, del)
    2. Welche bestehenden Funktionen benutze ich?
        - Ausnutzen was es schon gibt
        - Was für Argumente und Rückgabewerte werden dort benutzt?
    3. Habe ich Security-relevante oder Performance-kritische Datenübertragungen?
    4. Gibt es kapselbare / immer wiederkehrende / redundante HTML-Ausgaben?
        - Darstellen-Funktionen erstellen
5. **Datenbank anpassen** (meistens um Tabellen erweitern)
6. S**QL Dateien anpassen** (Security-Stufen beachten)
7. **Klassen definieren** (Klassenname_def.php Namenskonvention beachten)
8. **Install-Script anpassen** (Links nötig? - -> config_part4.sh)
9. **Includes anpassen** (für Objekte und weitere Module)
    - Neue Klassen werden je in USER_ARTIKEL_HANDLING.php und in USER_BESTELLUNG.php eingebunden. Wenn es eine reine Administrator Funktion ist, kann man sie auch erst in SHOP_ADMINISTRATION.php einbinden.
10. **Funktionen schreiben** (auch get, new, upd, del neben der eigentlichen Funktionalität)
11. **Funktionstests**
12. **Formularmasken programmieren** (inkl. JavaScript Eingabeprüfung und Sonderzeichen-Unterstützung)

 © 2001 R. Glanzmann & J. Fontanil

13. **Aufrufe in bestehende oder neue Module integrieren** (darstellen = ...)
14. **Eingliederung der neuen Funktionalität in die Shop oder Admin-Menüstruktur**
15. **Test auf Sonderzeichen und Extremwertetest** (negative, 0 und grosse positive Werte)
16. **Test der Gesamtfunktionalität**
17. **Commit auf SourceForge**

 © 2001 R. Glanzmann & J. Fontanil

1.8.8 Variablen zwischen den Webpages transportieren

Um einen Variablenwert von einer HTML-Seite zur Nächsten weiterzugeben, können folgende Verfahren verwendet werden:

Übermittlung direkt im Link (GET)

Bei der direkten Datenübertragung werden der URL die zusätzlich zu Übertragenen Variablennamen und Werte angehängt. Diese werden dann in der URI dem nächsten Dokument übergeben.

Aufbau der URI:

```
Link ? Variablenname_1 = Variablenwert_1 & Variablenname_2 =
Variablenwert_2 & ...
```

Praktisches Beispiel:

```
href="./auswertung.php?name=mueller&vorname=hans&darstellen=1"
```

Es ist zu beachten, dass die maximale URI-Länge von Browser zu Browser unterschiedlich ist. Es sollte keine URI übergeben werden, die länger als 2 KByte ist. Diese Längenbeschränkung grenzt das Einsatzgebiet stark ein. Ausserdem dürfen Sonderzeichen nicht direkt übergeben werden!

Wir haben dieses Verfahren während der ersten Projektarbeit einige Male dazu verwendet, die Variable für die Darstellen-Weiche weiterzugeben. Da der Surfer jedoch die ganze URI sieht und natürlich auch verändern kann, besteht die Gefahr, dass er unvorhersehbare und nicht erwünschte Zustände herstellen kann. Aus diesem Grund haben wir weitgehendst auf den weiteren Einsatz von GET verzichtet.

Übermittlung als Feld in einem Formular (POST oder GET)

Bei der Übermittlung der Variablen in einem Formular hat man die Wahl zwischen GET und POST. GET funktioniert, wie oben detailliert beschrieben, durch Anhängen der Daten an die URL. Es gilt hier dementsprechend auch die gleiche Längenbeschränkung (2 KByte). Sonderzeichen werden aber vom Browser ASCII-codiert übergeben und auch wieder decodiert.

Da beim Formularversand mittels GET die URI wiederum vom Browser angezeigt wird und natürlich auch manipuliert werden kann, ist von dieser Methode abzuraten.

Die Variablenweiterleitung per POST-Formular ist die von uns favorisierte Methode, welche wir mit Abstand auch am häufigsten eingesetzt haben. Das ganze Formular ist in seiner Grösse nicht beschränkt. Es können somit auch grössere Datenbestände, wie Bilder, weitergegeben werden. Ein weiterer positiver Aspekt ist, dass dies für den Surfer transparent geschieht. Er kann die Daten nicht so einfach manipulieren und sieht den übermittelten Datensatz nur im HTML-Quellcode.

Die Variablen können in allen möglichen Formularfeldern übermittelt werden:

Formularfeldtyp	*Beschreibung*
text	Einzeiliges Eingabefeld
checkbox	Kontrollkästchen
radio	Optionsschaltfläche
submit	Sendeknopf

 © 2001 R. Glanzmann & J. Fontanil

Formularfeldtyp	*Beschreibung*
reset	Knopf zum zurücksetzen des Formulars
textarea	Mehrzeiliges Texteingabefeld
select	Pull-Down-Menu
passwort	Verdecktes Eingabefeld
button	Schaltfläche
hidden	Unsichtbares Feld

Aufbau eines Formluarfeldes:

`<input type="`**`Formularfeldtyp`**`" name="`**`Variablenname`**`" value="` **`Variablenwert`**`">`

Beispiel eines Formulars mit einem text- und einem hidden-Feld:

```
<form method='post' action='./index.php'>
     <input type="hidden" name="darstellen" value="1">
     <br><B>Geben Sie ihre Land ein:</B>
     <input type="text" name="Land" size="30" value="Schweiz">
     <input type="submit" name="weiter" value="weiter">
</form>
```

Mit dem Attribut value kann den meisten dieser Eingabefelder ein Initialwert zugewiesen werden. Dieser wird dem Surfer dann angezeigt und kann editiert werden. Der editierte Wert wird dann beim Drücken des Sendeknopfs als Variable übermittelt.

Um eine Variable unverändert weiterzugeben (zum Beispiel über mehrere HTML-Seiten), kann sie in ein hidden-Feld geschrieben werden. Dieses kann vom Surfer nicht verändert werden und ist nur im Quellcode ersichtlich. Das hidden-Feld eignet sich auch sehr gut zur Weitergabe der Darstellungsweiche (Entscheidung, welche HTML Seite als nächstes ausgegeben wird).

Was unbedingt beachtet werde muss, ist, dass Formularfelder nach der Übermittlung bei gewissen Sonderzeichen (z.B. \,',") einen Backslash vorgestellt haben. Diese Backslashes müssen mit der PHP-Funktion `stripslashes` entfernt werden. Gibt man eine Variable zum Beispiel in einem Hidden-Feld dreimal weiter, ohne jeweils die `stripslashes` bei jedem Schritt auszuführen, so wird jedem Sonderzeichen bei jeder Weiterleitung ein Backslash vorangestellt, wobei dieser bei der nächsten Weiterleitung auch wieder als Sonderzeichen behandelt wird!

Folgendes Beispiel verdeutlicht, was passiert, wenn `stripslashes` vergessen wird:

Orginalstring:	`/sonderzeichen\ 'string'`
Einmal weitergegeben:	`/sonderzeichen\\ \'string\'`
Zweimal weitergegeben:	`/sonderzeichen\\\\ \\\'string\\\`
Dreimal weitergegeben:	`/sonderzeichen\\\\\\\\ \\\\\\\'string\\\\\\\'`

Welche Daten sollen wie übergeben werden?

Erst in einer späten Phase der Programmierung haben wir angefangen, die Formulare und deren Verarbeitung auf Herz und Nieren zu prüfen. Wir mussten jedoch schnell feststellen,

 © 2001 R. Glanzmann & J. Fontanil

dass zwar unsere logischen Überlegungen stimmten, es jedoch trotzdem möglich war, mit extremen Eingaben ein Chaos zu veranstalten.

In folgender Tabelle haben wir zusammengefasst, wie die Daten in den verschiedenen Fällen Codiert/Decodiert werden müssen, damit sie ordnungsgemäss ankommen.

Legende:

FORM:	Formularfeld; kein Unterschied zwischen sichtbar und hidden.
FORM (sichtbar):	sichtbares, vom Surfer veränderbares Formularfeld
FORM (hidden):	für den Surfer weder sichtbares, noch veränderbares Formularfeld
PHP:	PHP-Variable
DB:	Datenbank

was wird übertragen	*Von? Nach*	*Codier-/Decodier Massnahmen*
[a-z], [A-Z], [0-9]	FORM? PHP	keine
[a-z], [A-Z], [0-9]	FORM? DB	keine
[a-z], [A-Z], [0-9]	PHP? DB	keine
[a-z], [A-Z], [0-9]	PHP? FORM (sichtbar)	keine
[a-z], [A-Z], [0-9]	PHP? FORM (hidden)	keine
[a-z], [A-Z], [0-9]	DB? PHP	keine
Alle Zeichen inkl. Sonderzeichen	FORM (sichtbar)? PHP	`stripslashes`
Alle Zeichen inkl. Sonderzeichen	FORM (hidden)? PHP	1. `urldecode` 2. `addslashes`
Alle Zeichen inkl. Sonderzeichen	FORM (sichtbar)? DB	keine
Alle Zeichen inkl. Sonderzeichen	FORM (hidden)? DB	1. `urldecode` 2. `addslashes`
Alle Zeichen inkl. Sonderzeichen	PHP? DB	`addslashes`
Alle Zeichen inkl. Sonderzeichen	PHP? FORM (sichtbar)	`htmlspecialchars`
Alle Zeichen inkl. Sonderzeichen	PHP? FORM (hidden)	`urlencode`
Alle Zeichen inkl. Sonderzeichen	DB? PHP	keine

Der Weg vom Formular zur Datenbank geht eigentlich über eine PHP-Variable. Da aber die Daten vom Formular her schon mit vorangestellten Backslashes kommen, kann das stripslashes/addslashes-Prozedere entfallen, weil die Daten sowieso mit vorangestellten Backslashes an die Datenbank übergeben werden müssen.

PHP-Funktionen: `addslashes` und `stripslashes`

Mit der Funktion `addslashes` werden Sonderzeichen behandelt, die oft von Datenbanken interpretiert werden.

' " \ NUL

NUL steht dabei für den absoluten Wert 0. Diesen Sonderzeichen wird ein Backslash voran-

 © 2001 R. Glanzmann & J. Fontanil

gestellt, damit sie nicht interpretiert werden. Die Funktion `stripslashes` macht die durch `addslashes` vorgenommenen Umwandlungen wieder rückgängig. Der Vorgang wird übrigens auch oft als „escapen“ bezeichnet.

Werden Formulardaten übergeben, bei denen sich eines oder mehrere der oben aufgelisteten Sonderzeichen im String befinden, fügt der Browser automatisch Escape-Zeichen ein,

Eingabesting	*Funktion*	*Ausgabestring*
"string" zum 'übertragen'	`addslashes`	\"string\" zum \'übertragen\'
\"string\" zum \'übertragen\'	`stripslashes`	"string" zum 'übertragen'

PHP-Funktionen: `urlencode` und `urldecode`

Die beiden Funktionen urlencode und urldecode sind eigentlich dazu entwickelt worden, Strings, die direkt an die URL-angehängt werden, so umzuwandeln, dass sie den Konventionen für den Aufbau einer URL genügen.

Dies geschieht, indem Zeichen, welche nicht direkt in der URI übergeben werden können, hexadezimal codiert werden. Dabei handelt es sich um die Zeichenfolge &HH, wobei HH für den hexadezimalen ASCII-Wert des speziellen Zeichens steht. Leerschläge werden durch ein '+' ersetzt.

Da URL-encodierte Strings weder Sonderzeichen, die escaped werden könnten, noch Leerschläge beinhalten, eignet sich diese Codierung hervorragend zur unkomplizierten Weitergabe von Daten über mehrere Seiten in Hidden-Feldern.

Enthält ein String viele Sonderzeichen, kann dessen URL-encodierter String viel länger als der Orginalstring werden, da jedes Sonderzeichen durch drei Codier-Zeichen ersetzt wird. Dies muss vor allem beachtet werden, wenn an irgend einer Stelle eine Längenbeschränkung verwendet wird.

Eingabesting	*Funktion*	*Ausgabestring*
"string" zum 'übertragen'	`urlencode`	%22string%22+zum+%27%FCbertragen%27
%22string%22+zum+%27%FCbertragen%27	`urldecode`	"string" zum 'übertragen'

Zu beachten ist, dass der Punkt von urlencode nicht als Sonderzeichen behandelt wird, was zu Problemen führen kann, wenn ein Variablenwert als Variablenname verwendet werden soll. Mit der Funktion str_replace kann auch der Punkt noch durch die entsprechende ASCII-Sequenz (%2E) ersetzt werden.

```
str_replace(".", "%2E", $einsetzen);)
```

`urldecode` ersetzt alle ASCII-Sequenzen mit dem entsprechenden Zeichen, wodurch eine gesonderte Behandlung des Punktes wegfällt.

PHP-Funktion: htmlspecialchars

Die Funktion htmlspecialchars wandelt die folgenden Zeichen so um, dass die Darstellung in HTML korrekt erfolgt:

 © 2001 R. Glanzmann & J. Fontanil

- & wird als & dargestellt
- " wird als " zurückgegeben
- < wird in < umgewandelt
- > wird als > ausgegeben

Eingabesting	*Funktion*	*Ausgabestring*
"string" <zum> &übertragen&	urlencode	"string" <zum> &übertragen&

HTML-Ausgaben müssen unbedingt mit urlencode codiert werden, wenn man verhindern will, dass Zeichenfolgen, die sich in eckigen Klammern (<>) befinden, als Tag interpretiert werden. Wir nutzten diese Funktion vor allem, um Forumularfeldwerte mittels value-Attribut auszugeben:

```
<?php
     $variable='enthält "HTML"-Sonderzeichen und <Tags>';
?>

<form method='post' action='./index.php'>
     <br><b>Editieren Sie den Text:</b>
     <input type="text" name="form_data"
            value="<?= htmlspecialchars($variable)" size="40">
     <input type="submit" name="weiter" value="weiter">
</form>
```

1.8.9 JavaScript

Ursprünglich war die Verwendung von JavaScript im Shop nicht geplant. Durch der Einführung von Unterkategorien standen wir plötzlich vor dem Problem, dass wegen des dynamischen Menüs im linken Frame mit einem Mausklick zwei Frames aktualisiert werden mussten. Da dies mit reinem HTML nicht möglich ist, standen wir vor der Wahl, entweder den ganzen Shop so umzuschreiben, dass auf Frames verzichtet werden kann oder eine JavaScript Funktion aufzurufen, die beide Frames gleichzeitig aktualisiert. Wir entschieden uns für letztere Lösung, da sie wesentlich weniger Zeitaufwand zur Implementation benötigte.

Gleichzeitige Aktualisierung von Frames mittels JavaScript

Folgende kleine JavaScript Funktion wird dazu benötigt:

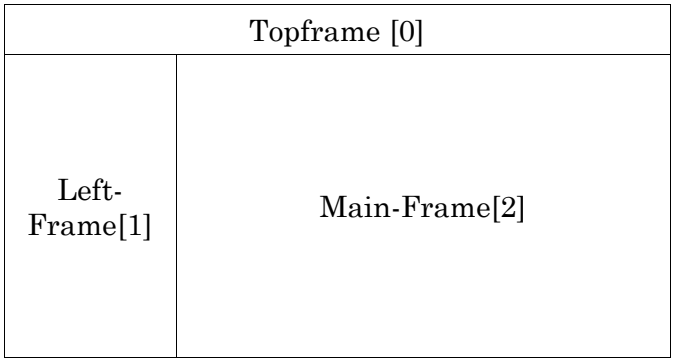
```
<script language="JavaScript">
     function zweiframes(url1,f1,url2,f2) {
     parent.frames[f1].location.href=url1;
     parent.frames[f2].location.href=url2;
     }
</script>
```

Funktionsargumente:

`url1, url2:` URL des Dokuments, welches aufgerufen werden soll

`f1, f2`: Nummer des zu aktualisierenden Frames

 © 2001 R. Glanzmann & J. Fontanil



Die Funktion „zweiframes“ wird direkt im Link folgendermassen aufgerufen:

```
<a href='javascript:zweiframes("url1", "f1", "url2", "f2")'>Link-
text</a>
```

Überprüfung eines Formularfeldes auf Eingabe

Um im Shop einkaufen zu können, muss nun fortwährend im Browser JavaScript eingeschaltet sein. Diesen Umstand nutzten wir gleich dazu, auch noch die Eingabeüberprüfung der Formulardaten per JavaScript zu tätigen. JavaScript hat im Gegensatz zu PHP den Vorteil, dass die Überprüfungsfunktion gleich mit dem Formular selbst zum Browser gesendet werden kann. Die Überprüfung, ob alle Felder ordnungsgemäss ausgefüllt wurden, geschieht demzufolge beim Surfer (Client-seitig) und erfordert weder zusätzlichen Datenverkehr, noch Rechenpower vom Server. Gibt ein Surfer zum Beispiel die Daten fünfmal nicht ordnungsgemäss ein, wird dadurch kein einziges Byte zum Server gesendet. Macht man diese Kontrolle mittels PHP, müssen neun (fünf Fehlermeldungs- und vier Formular-) HTML-Seiten vom Server zum Client geschickt werden, was einem massiven Mehraufwand an Ressourcen entspricht.

Folgendes Beispiel illustriert die minimalste Art der Eingabeüberprüfung. Es wird nur überprüft, ob der Surfer etwas in das Formularfeld eingegeben hat:

```
<script language="JavaScript">
     function chkFormular() {
          /* Überprüfung, ob kein Leerstring übergeben wurde */
          if(document.Formular.Vorname.value == "") ) {
          /* Alert-Warnung ausgeben, wenn ein Leerstring übergeben
             wurde */
          alert("Bitte einen Vornamen eingeben!");
          /* den Fokus auf das Feld Vorname setzen */
          document.Formular.Vorname.focus();
          /* false zurückgegeben, damit das Formular nicht abge
             schickt wird /*
          return false;
     }
</script>

<form name="Formular" onSubmit="return chkFormular()" method='post'
  action= './index.php' >
     <br><b>Geben Sie Ihren Vornamen ein:</b>
     <input type="text" name="Vorname" size="40">
     <input type="submit" name="weiter" value="weiter">
```

 © 2001 R. Glanzmann & J. Fontanil

```
</form>
```

Hat man viele Felder zum Überprüfen in einem Formular, lohnt es sich, die Überprüfung der Felder in einer for-Schleife durchzuführen, da sonst eine nicht vernachlässigbare Menge an JavaScript Code übermittelt werden muss.

Unsere Meinung zu JavaScript

Wir haben das Programmierprinzip von JavaScript als sehr angenehm empfunden. Dank der konsequenten Objektorientiertheit dieser Scriptsprache, war es problemlos möglich, an alle gewünschten Webelemente zu kommen.

Dies bewog uns dann dazu, JavaScript auch im Hauptmenü des Kategorienmanagements zu verwenden. Leider machte uns hier der Netscape 4.7-Browser wieder einmal einen Strich durch die Rechnung, da er ein schon ausgegebenes Bild nicht durch ein anderes ersetzen kann, wenn es nicht die exakt gleichen Dimensionen besitzt. Wir bauten eine Browserweiche ein und gaben fortan Netscape 4.7-Browsern ein eigenes Menu aus.

Was wir bei der Entwicklung von JavaScript Funktionen als äusserst störend empfanden, war, dass man vom Browser nur mit äusserst spärlichen Informationen versorgt wird, wenn ein Problem mit einer JavaScript Funktion vorliegt. Es ist vorgekommen, dass mitten in der Funktion die Ausführung ohne Fehlermeldung oder Warnung abgebrochen wurde. Etwa zwei Wochen vor dem Ende der Diplomarbeit ist dann Mozilla 0.9.5 freigegeben worden. Dieser Browser enthält eine sehr nützliche JavaScript-Konsole und sogar einen JavaScript-Debugger. Leider mussten wir bis zu diesem Zeitpunkt ohne ein solch komfortables Tool auskommen.

 © 2001 R. Glanzmann & J. Fontanil

1.8.10 Datenbankdesign

Unterkategorien

Auch im Datenbankdesign haben wir Erfahrungen gesammelt. Als es darum ging Unterkategorien zu implementieren, haben wir uns nach intensivem Gedankenaustausch dafür entschieden, die Unterkategorie-Tiefe auf maximal eine zu beschränken. Dies vor allem, weil in einem Shop im linken Frame die Darstellung der Auswahl-Kategorien mit mehreren Verschachtelungen ungewohnt aussieht und sich nicht intuitiv bedienen lässt, wenn man immer nur einen offenen Kategoriebaum darstellen will (aus Übersicht). Weiter würde mit tieferen Verschachtelungen die ohnehin schon beschränkte Namenslänge weiter eingeschränkt werden (sichtbarer Bereich).

Die einfachste und zeitsparendste Implementationsmethode ist, die Kategorientabelle um das Attribut 'Unterkategorie_von' zu erweitern. Dieser Designentscheid ist unkonventionell und aus Sicht der Datenbank nicht unbedingt optimal. Die Datenbank bildet hier die logische Verknüpfung von Kategorie und ihren Unterkategorien nicht ab. Um die logische Verknüpfung nachzubilden, hätte man eine Tabelle Unterkategorien einfügen müssen und diese mit einer '1 zu n'-Beziehung an die Kategorien-Tabelle hängen müssen.

Wir haben uns hier für den einfacheren Weg entschieden, im Bewusstsein, dass die Datenbank unser Datenmodell nicht genau nachbildet. Wir können dies folgendermassen begründen:

- Zeitersparnis
- Mehrere Unterkategorie-Ebenen sind bei Bedarf immer noch möglich, indem man im Unterkategorie_Attribut eine weitere Unterkategorie angibt. Man ist also nicht definitiv auf nur eine Ebene beschränkt.
- Viele Funktionen konnten unverändert mit der erweiterten Tabelle arbeiten
- Keine weiteren Klassen mussten eingeführt und included werden

Ein Designfehler ist uns aber dennoch unterlaufen: Wir haben nicht weit genug in die Zukunft geschaut. Im Attribut Unterkategorie_von, setzten wir den *Namen* der Haupt-Kategorie ein. Wie wir später erfahren mussten ein in mehrfacher Hinsicht schwerwiegender Fehler, welcher uns aber neben den zu umschiffenden Unwägbarkeiten auch einiges an Erfahrung einbrachte.

Wann immer wir performance-kritische Funktionen zu schreiben hatten, übergaben wir die benötigten Attribute in einem Formular als hidden-Field oder per Link im Query-String. Hier kamen nun allerlei Encodierungsprobleme auf uns zu, da der Shop auch Sonderzeichen richtig behandeln können sollte (siehe auch Kapitel 1.8.8). Weiter waren Kategorienamen vor der Implementierung von Unterkategorien eindeutig und einmalig (wir hatten für den Kategorienamen sogar ein UNIQUE-Statement in der Datenbank). Auf einmal kann man aber mehrere Kategorien mit gleichem Namen haben. Da wir hier nicht mit der Kategorie-ID arbeiten, erbt die neue Kategorie alle Unterkategorien der schon bestehenden. Solche Probleme hätten sich vermeiden lassen, wenn man sich auf die Kategorie-ID bezogen hätte.

Wir haben daraus gelernt und werden wohl nie mehr ein anderes Verknüpfungsattribut als eine ID wählen.

Shop-Settings

Die Shop-Settings Tabelle beinhaltet die allgemeinen Shopeinstellungen. Dies sind inzwischen ganze 40 Attribute, welche auf einer Zeile gespeichert werden. Wir haben uns in der Praktischen Arbeit gedacht, dass man pro Shop eine Zeile in dieser Tabelle hat und auf diese

 © 2001 R. Glanzmann & J. Fontanil

Weise bequem mehrere Shops in dieser Tabelle verwalten kann. Heute wissen wir, dass das so nicht funktioniert. Die parallel laufenden Shops haben alle eine eigene Datenbank. Die Shop-Settings-Tabelle wird zunehmend zu einem Problem, weil die eine Zeile viel zu lang ist. Unser Dozent hat uns darauf hingewiesen, dass z.B. die Datenbank Sybase und auch andere Datenbanken eher restriktiv sind, was ihre maximale Zeilenlänge anbelangt. In nächster Zukunft müsste man also diese Tabelle umstrukturieren, zum Beispiel so:

- Attributname varchar(256);
- Attributwert varchar(256);

Dies bedingt aber auch ein Anpassen vieler Funktionen und wurde deshalb auf die Zeit nach der Diplomarbeit verschoben. Es handelt sich hier um reine Fleissarbeit ohne weiteren Lerneffekt.

 © 2001 R. Glanzmann & J. Fontanil

1.9 Zukunft des PhPepperShops

Da wir nicht wussten, wie lange wir den Testserver der Schule noch zur Verfügung haben, registrierten wir die Domain **phpeppershop.com**. Diese Domain ist seit dem 13.09.2001 aufgeschaltet und wird seither als offizielle Informationsseite betrieben.

phpeppershop.com - Herzlich Willkommen auf www.phpeppershop.com - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Verlauf
Adresse http://www.phpeppershop.com
Wechseln zu Links
PhPepperShop
Sie befinden sich auf der Website des in Entwicklung stehenden Open-Source Webshopsystems PhPepperShop. Dieser Shop ist während einer Projektarbeit der Zürcher Hochschule Winterthur unter der Bezeichnung "Framework für einen PHP-Webshop" entstanden und wird momentan im Rahmen unserer Diplomarbeit zum PhPepperShop weiterentwickelt.
Auf folgenden Seiten finden Sie weitere Informationen:
Beispielshop
Homepage "Framework für einen PHP-Webshop"
SOURCE FORGE
ENGLISH
Falls Sie Fragen, Anregungen oder Bemerkungen an uns haben, können Sie uns unter folgender EMail-Adresse erreichen: developers@phpeppershop.com
Letzte Aktualisierung: 14.09.2001
website powered by
glaro
Fertig
Internet

Mit dem Abschluss des Studiums werden wir auch unsere bis anhin verwendeten ZHW E-Mail-Adressen nicht mehr zur Verfügung haben. Damit wir auch danach noch für Fragen und Feedback erreichbar sein werden, richteten wir eine offizielle Kontaktadresse ein:

developers@phpeppershop.com

1.9.1 Veröffentlichung des Final-Release

Als offizielles Veröffentlichungsdatum haben wir den **15. November 2001** festgesetzt. Bis zu diesem Datum müssen jedoch noch einige Arbeiten erledigt werden:

- Backup des Shops automatisieren
- Bugfixes
- FAQ-Liste und User-Forum auf http://www.phpeppershop.com einrichten

1.9.2 Leute, die den Shop weiterentwickeln wollen

Folgende Leute haben sich bereit erklärt, unter Umständen an der Weiterentwicklung des Shops mitzuarbeiten, oder den Shop in eine andere Sprache zu übersetzen:

- Uli und Christian / deutsch / Andalusien?? wollen mitentwickeln
  Uli@prismaworld.com oder Webmaster@fileload.de
- Hans, Physiker / englisch /? Security testen
  thesis@xs4all.nl
- Peter Fitzpatrick / englisch /? ?Secure Online Transaction (ihm Infos schicken)
  petefitzy@btinternet.com

 © 2001 R. Glanzmann & J. Fontanil

- Hans-Peter Ambros (Fan von pgmarket) / deutsch /? will mitentwickeln (v.a. spezialisierte Lösung)
  ambros@planet-interkom.de
- Hans-Jakob Martin / deutsch /? will mitentwickeln, falls er bis dann nicht schon einen anderen Shop benutzt (Lager, Buchhaltung)
  Hans-Jakob.Martin@gmx.net www.nordakademie.de
- Georg Spang / deutsch /? ?will mitentwickeln (spezialisierte Lösung)
  georg_spang@gmx.de
- Alex Henriksen / englisch /? ?übersetzt englischen Shop ins Dänische
  alexie@zbox.dk
- Steven Dowd / englisch /? ?übersetzt den Shop ins Englische
  steven.dowd@dowd.co.uk
- Phil / deutsch /? ?will Shop in PHP-Nuke integrieren
  p-fs@gmx.de
- kingluui / deutsch / ? ?Shop vom Deutschen ins Spanische übersetzen
  xxxforumbanner@kingluui.de

 © 2001 R. Glanzmann & J. Fontanil

1.10 Schlusswort

Wir haben unser Ziel erreicht! Der Shop kann der Öffentlichkeit übergeben werden. Wir wissen, dass er auch in Zukunft weiter entwickelt werden wird und freuen uns schon auf weitere interessante Features und Feedbacks.

Der PhPepperShop ist etwas Einmaliges. Er bietet eine enorme Flexibilität und Dynamik, welche auch kommerzielle und teure Shop-Lösungen oft vermissen lassen.

Aufgrund des enormen Echos auf unsere Projektarbeit von diesem Frühjahr, waren wir hoch motiviert, den Shop auf die Bedürfnisse des Marktes anzupassen. Dies ist uns unserer Meinung nach auch gelungen.

Bei Herrn Feisthammel möchten wir uns ganz herzlich für seine ausserordentliche Unterstützung, die er uns auch in diesem Projekt wieder gewährleistete, bedanken.

Speziellen Dank möchten wir unseren Freundinnen aussprechen, die uns auch in dieser letzten grossen Arbeit unseres Studiums viel Verständnis und Unterstützung entgegen brachten.

Winterthur, den 28.10.2001

Reto Glanzmann José Fontanil

 © 2001 R. Glanzmann & J. Fontanil

2. Architektur

2.1 Security

Wir haben uns viele Gedanken betreffend der Security des Shops gemacht und Sicherheitsmassnahmen auf mehreren Ebenen eingebunden. Im Folgenden gehen wir genauer auf die einzelnen Teile der Shop-Security ein.

2.1.1 PHP

Da ein potentieller Hacker/Cracker den Sourcecode der HTML-Seiten sowie Header-Informationen begutachten und verändern kann, besteht die Gefahr, dass er sich so ein Bild über den Shop-Aufbau machen kann und auf diese Weise zu wichtigen Hintergrundinformationen kommen könnte.

Um dem vorzubeugen, haben wir uns in der ersten Projektarbeit schon beim Design dafür entschieden, diejenigen PHP-Scripte, welche Administrations-Aufgaben wahrnehmen, von den USER-Level-Scripten so gut es geht zu entkoppeln.

Dies fängt schon bei den SQL-Kommandos an: Alle SQL-Befehle, die von USER-Funktionen benutzt werden, kommen aus einer Datei. Alle Administrator-Scripts befinden sich in einer anderen Datei, die von keinem USER-Modul included wird. Auf diese Weise ist sichergestellt, dass von einer USER-Funktion niemals ein Administrations-Script aufgerufen wird.

PHP bietet einen sogenannten `SafeMode` an. Durch ihn versucht PHP einige Sicherheitslücken zu schliessen. Dieser Modus beinhaltet ein Sicherheitskonzept, welches die Zugriffsberechtigung auf Dateien regelt. Ein Zugriff auf eine Datei wird nur noch gewährt, wenn die Datei, das Verzeichnis und das Script dem gleichen Benutzer gehören. Dieses Kriterium gilt weiter für alle Funktionen, die ein Risiko für die Systemsicherheit darstellen könnten. Weiter kann man ausführbare Dateien nur ausführen, wenn sie sich im save_mode_exec-Verzeichnis befinden. Dieses kann in der Datei php.ini definiert werden. Wir haben den `SafeMode` selbstverständlich eingeschaltet.

Damit ein Benutzer nicht eine Bestellung mit manipulierten Preisen aufgeben kann, werden diese immer aus der Datenbank geholt und an das jeweils anfordernde Script übergeben.

2.1.2 htaccess

Der Administrations-Teil wird weiter per htaccess geschützt. Alle Administrations-Scripte sind im Unterverzeichnis `Admin` abgelegt. Wer einen anderen Server (z.B. IIS) einsetzt, muss dafür sorgen, dass das nur vom Administrator auf das Verzeichnis `Admin` zugegriffen werden kann.

2.1.3 MySQL

Um dieses Konzept auch auf der Datenbank-Ebene fortzusetzen, arbeitet der Shop grundsätzlich mit zwei Datenbank Benutzern. Einerseits mit dem `shopuser` und um die Administrator-Scripte auszuführen als `shopadmin`. Dem `shopuser` wurden nur die nötigsten Rechte gewährt. Er kann den Grossteil der Datenbank nur lesen. Bestellungen darf er updaten und löschen.

MySQL selbst wird auch nicht als `root` gestartet, sondern mittels der Anweisung:

```
../mysql/bin/safe_mysqld --user mysql&
```

 © 2001 R. Glanzmann & J. Fontanil

Auf diese Weise wird MySQL von einem User gestartet, welcher analog zum Apache-User nur über sehr eingeschränkte Rechte im UNIX Betriebssystem verfügt.

Zum Schluss stellten wir noch sicher, dass die beiden User `shopuser` und `shopadmin` keine GRANT- und ALTER-Rechte haben.

In dem von vielen Personen gewünschten Ein-User-Betrieb des Shops, greifen die MySQL-Security Überlegungen nicht. Der Shop erledigt alle Operationen mit dem `shopdamin`-User.

Man kann den PhPepperShop auch mit einem (oder zwei) schon bestehenden Datenbank-Usern betreiben. Hier wissen wir natürlich nicht, welche Rechte diesen Usern gegeben wurden. Wir haben diese Flexibilität eingebaut, weil es viele Anwender gibt, welche an ihrem vom Hosting-Provider erhaltenen MySQL-Account nichts ändern dürfen. Man sollte sich aber im Klaren darüber sein, dass, wenn diesen Benutzern GRANT-/ALTER-Rechte oder sogar Rechte an weiteren Datenbanken gewährt werden, dies eine potentielle Sicherheitslücke darstellt.

2.1.4 SSL

Da wir im PhPepperShop Kreditkartenhandling eingeführt haben, müssen diese sensiblen Daten auch verschlüsselt übertragen werden. Dafür bietet sich SSL an. Wir haben SSL auf unserem Webserver installiert und zu Testzwecken mit einem selbst erzeugten Serverzertifikat betrieben. Damit ein Shopbenutzer dem Shop aber vertrauen kann, sollte man hier ein offizielles Serverzertifikat, z.B. von VeriSign, benutzen.

Der Shopadministrator kann in den allgemeinen Shopeinstellungen SSL optional einschalten. Es ist ihm überlassen, ob er SSL auf dem Webserver installieren will oder nicht. Wir empfehlen die Verwendung von SSL, sobald jemand Kreditkarten als Zahlungsmöglichkeit anbieten will. Man muss sich aber auch bewusst sein, dass SSL-Verbindungen 'langsam' sind.

Es wird beim PhPepperShop nicht der komplette Frameset über HTTPS übertragen, sondern nur der Content-Frame (welcher die Formulare enthält). Aus diesem Grund zeigen Netscape 4.7x und der Internet Explorer auch kein 'sicheres Schlösschen' an. Dies ist verwirrend. Wenn sich jemand überzeugen will, ob seine Eingaben wirklich SSL geschützt sind, so kann er dies folgendermassen tun:

- Mozilla / NS 6.x: Zeigt auch hier das 'Sicherheits-Schlösschen' an.
- Internet Explorer: Rechts-Klick -> Eigenschaften -> dort muss https als Protokoll stehen.
- Netscape 4.7x: Rechts-Klick -> Informationen anzeigen -> dort muss https als Proto koll stehen.

2.1.5 GNU-PG

Eurocard und VISA bieten einen lukrativen MailOrder Vertrag an, sobald man nachweisen kann, dass die Kreditkartendaten der Kunden nie unverschlüsselt übertragen werden. Damit dies gewährleistet ist, bietet sich eine Kombination aus SSL und GNU-PG an. Die Kreditkartendaten werden per SSL-verschlüsselt vom Kunden an den Shop übermittelt. Das Bestellungs-E-Mail für den Shopadministrator wird dann vor dem Versand mit GNU-PG oder PGP verschlüsselt. Da die Kreditkartendaten nicht in die Shop-Datenbank gespeichert werden, können sie nie unverschlüsselt abgerufen werden. Nur der Shopbetreiber kann sie mit seinem privaten Schlüssel wieder entschlüsseln.

Aufgrund, der in den Erfahrungen erwähnten Problemen, ist die E-Mail Verschlüsselung mittels GNU-PG leider noch nicht implementiert worden.

 © 2001 R. Glanzmann & J. Fontanil

2.1.6 Kunden-ID

Damit die Kunden identifiziert werden können, erhält jeder eine Kunden_ID. Diese wird über gewisse Formulare hinweg als hidden-Field übertragen. Damit hier nicht ein Sicherheitsloch entsteht, haben wir die Kunden-ID nicht einfach als fortlaufende Integer Zahl implementiert. Sie besteht aus einer Ganzzahl, welche folgendermassen zusammengesetzt ist:

1. Mit dem PHP-Zufallszahlengenerator `rand( )` erzeugen wir drei unabhängige Zufallszahlen aus dem Bereich von 1 bis RAND_MAX (bei uns 2'147'483'647)
2. Diese drei Zahlen werden als String aneinander gereiht und auf ihre Einmaligkeit unter den schon bestehenden Kunden_IDs hin überprüft. Es entsteht ein Zahlenraum von 111 bis 214'748'364'721'474'836'472'147'483'647. Die Kunden_ID in der Tabelle `kunde` hat einen eigenen Index und wird als `varchar(255)` gespeichert.
3. Ist die Zahl einmalig, so erhält der Kunde diese Kunden-ID.

Mit dieser Kunden-ID wird es böswilligen Personen praktisch verunmöglicht, durch Raten von Kunden-IDs an Kunden-Informationen zu gelangen, respektive unter deren Namen zu bestellen.

 © 2001 R. Glanzmann & J. Fontanil

2.2 PHP-Module und -Funktionen

Im folgenden wird nun die Implementation des Shops aus struktureller Sicht genauer erläutert. Die Übersicht soll helfen, den Aufbau des Shops besser verstehen zu können. Die Modul- und Funktionsbeschreibungen geben einen Überblick über die Abbildung der Shop-Funktionalitäten auf die PHP-Scripte. Damit man den Unterschied von der Projektarbeit zur Diplomarbeit sieht, wurden geänderte und neu geschriebene Funktionen und Module mit einem **DA** gekennzeichnet.

2.2.1 Übersicht

Der Shop wurde funktionell in mehrere Schichten unterteilt. Die PHP-Module (Dateien) sind jeweils einer der Schichten zugeordnet.

Schichtenmodell des PhPepperShops
HTML-Darstellung (Frameset, Kunden-Navigation im top-Frame)
Darstellen Module (hier werden die komplexen Abläufe in Funktionen mit HTML / Plain-Text Output behandelt)
Aufruf Module (rufen Funktionen auf und generieren HTML)
Payment-Interface (Schnittstelle für externes Zahlungssystem)
PHP-Funktionen
Datenbankanbindung (Getrennte User- und Administrator Datenbankanbindung)
Datenbank Wrapper (Spricht verschiedene Datenbanken an, bietet nach oben eine einheitliche Schnittstelle für Datenbank-Funktionen)
SQL-Statements (User- und Administrator SQL-Statements zentral in je einer Datei)
Klassendefinitionen (Definitionen der komplexeren Datenstrukturen)

Es folgt nun eine Einteilung der entsprechenden Module in ihre Schichten. Weiter ist jedem einzelnen Modul eine Sicherheitsstufe zugeordnet. User Funktionen arbeiten mit User SQLs und User Datenbank Objekten. Die Includes sind so gehalten, dass kein User-Modul auf irgendwelche als Admin deklarierte Module zugreifen kann. Admin Module dürfen hingegen auch User-Module benutzen. Die Pfadangaben beziehen sich, wenn nicht anders angegeben, auf folgenden Pfad:

```
<shopdir>/shop/
```

 © 2001 R. Glanzmann & J. Fontanil

Schicht	Beschreibung	Modul-Name	Security
HTML-Darstellung	Frameset	../index.php	User
	Oberer Frame	Frameset/top.php	User
Darstellen Module	Bestellung	USER_BESTELLUNG_DARSTELLUNG.php	User
	User Hilfe darstellen	USER_ADMIN_HILFE.php	User
	Bild aus DB anzeigen	bild_view.php	User
	Artikelmanagement	Admin/SHOP_ADMINISTRATION_ARTIKEL.php	Admin
	Admin Hilfe darstellen	Admin/ADMIN_HILFE.php	Admin
	Erstellt ein DB-Backup	Admin/ADMIN_backup.php	Admin
	Zurücklesen von Backup	Admin/ADMIN_restore.php	Admin
	Externe Zahlungsstelle	payment_interface.php	Admin
Aufruf Module	Artikelhandling	USER_ARTIKEL_HANDLING_AUFRUF.php	User
	Bestellung	USER_BESTELLUNG_AUFRUF.php	User
	Bestellung	USER_BESTELLUNG_1.php	User
	User PopUp anzeigen	pop_up.php	User
	Administration	Admin/SHOP_ADMINISTRATION_AUFRUF.php	Admin
	Backup/Restore Menü	Admin/SHOP_BACKUP.php	Admin
	DB-Backup des Shops	Admin/SHOP_BACKUP_f1.php	Admin
	Bestellungsmanagement	Admin/SHOP_BESTELLUNG.php	Admin
	Kundenattribute	Admin/SHOP_KUNDE.php	Admin
	Layout Management	Admin/SHOP_LAYOUT.php	Admin
	Shop Einstellungen	Admin/SHOP_VERSANDKOSTEN.php	Admin
	Versandkosten	Admin/SHOP_SETTINGS.php	Admin
	Kategorienmanagement	Admin/Shop_Einstellungen_Menu_Kategorien.php	Admin
	Administrationsmenü	Admin/Shop_Einstellungen_Menu_1.php	Admin
	Artikelmanagement	Admin/bild_up.php	Admin
	Admin PopUp anzeigen	Admin/pop_up_admin	Admin
Payment-Interface	externes Payment	payment_interface.php	User
	externes Payment/Test	pay_ext_test.php	User
PHP-Funktionen	Artikelhandling	USER_ARTIKEL_HANDLING.php	User
	Bestellung	USER_BESTELLUNG.php	User
	Administration	Admin/SHOP_ADMINISTRATION.php	Admin
SQL-Statements	User SQL-Statements	USER_SQL_BEFEHLE.php	User
	Admin SQL-Statements	Admin/ADMIN_SQL_BEFEHLE.php	Admin
Datenbank-anbindung	User DB Anbindung	initialize.php	User
	Admin DB Anbindung	Admin/ADMIN_initialize.php	Admin
Datenbank Wrapper	DB Kommunikation	database.php	User

© 2001 R. Glanzmann & J. Fontanil

Schicht	Beschreibung	Modul-Name	Security
Klassendefinition	Artikel-Klasse	artikel_def.php	User
	Attribut-Klasse	attribut_def.php	User
	Bestellungs-Klasse	bestellung_def.php	User
	Kategorien-Klasse	kategorie_def.php	User
	Kreditkarten-Klasse	kreditkarte_def.php	User
	Kunden-Klasse	kunde_def.php	User
	Versandkosten-Klasse	versandkosten_def.php	User

Alle weiteren Dateien wurden nicht ins PHP-Schichten-Modell integriert, haben aber eine Security Stufe.

Beschreibung	Modul-Name	Security
User Redirect	../index.html	User
Administrationtool Redirect	Admin/index.html	Admin
Datenbank und DB-User erzeugen	database/{shopname}_create.sql	Admin
Datenbank mit Default-Shop füllen	database/{shopname}_insert.sql	Admin
Datenbank und DB-User löschen	database/{shopname}_del.sql	Admin
Installation (Diese Dateien existieren nur vor der Installation)	../config.pl	Admin
	../config_part4.sh	Admin
Deinstallation	../remove.pl	Admin
CSS-Template	Admin/csstemplate.txt	Admin
Frameset Template	Admin/indextemplate.txt	Admin
User CSS Datei	shopstyles.css	User
User CSS Datei (doppelt wegen NS. 4.7x)	Frameset/shopstyles.css	User
Fixes Admin CSS-Layout	Admin/shopstyles.css	Admin
Willkommenspage	Frameset/content.html	User

2.2.2 PHP-Module und ihre Funktionen

Es folgen nun die nach Schichten sortierten PHP-Module. Handelt es sich um Aufruf-Module, so wird deren Funktion anhand der übergebenen Variable 'darstellen' definiert. Bei den funktional gegliederten Modulen werden die Funktionen erklärt.

Klassendefinitionen

Wir haben im PhPepperShop immer auf Klassen zurückgegriffen, wenn komplexe Datentypen bearbeitet oder übertragen (von Funktion zu Funktion) werden mussten. Der PhPepperShop ist ganz klar nicht Objekt-orientiert entwickelt. Wir benötigten die Klassen lediglich als Daten-Container.

 © 2001 R. Glanzmann & J. Fontanil

Modul: Klassendefinitionen	
Datei	**Klassenname**
artikel_def.php	Artikel
Beschreibung **DA**	Definiert eine Klasse Artikel, um mit den vielen Attributen eines Artikels umgehen zu können. Ein Artikel-Objekt kann auch alle Optionen und Variationen (beliebige Menge) eines Artikels fassen. Auf die Variablen der Klasse wird direkt, ohne set- / get-Funktionen zugegriffen. Für die Optionen- und Variationen-Arrays gibt es entsprechende Funktionen.
artikel_def.php	Option und Variation
Beschreibung **DA**	Im PhPepperShop werden Artikel Optionen und -Variationen in Arrays transportiert. Zukünftige Entwicklungen für den PhPepperShop sollen aber diese beiden Klassen verwenden. Sie bilden eine Zeile der jeweiligen Tabelle (artikel_optionen, artikel_variationen) ab. Man hat auf diese Weise viel elegantere Zugriffsmöglichkeiten auf die Daten.
artikel_def.php	Artikelmitkategorien
Beschreibung **DA**	Diese etwas seltsam benannte Klasse wurde für den Transport von Artikeln und ihren Kategorien erstellt. Seitdem ein Artikel in mehreren Kategorien gleichzeitig sein kann, musste man einen Weg haben, beide Informationen auf einmal transportieren zu können. Diese Klasse fasst deshalb ein Artikel-Objekt und einen Array von Kategorien.
attribut_def.php	Attribut
Beschreibung **DA**	Definiert die Klasse Attribut. Es existiert eine Tabelle 'attribut', in welcher diese Attribute gespeichert werden. Die Attribute werden dazu verwendet, um dynamisch Kundendatenfelder konfigurieren zu können. Man kann z.B. das Eingabefeld Strasse erzeugen und diesem (Kunden-)Attribut dann mitteilen, ob es angezeigt werden soll, ob seine Eingabe überprüft werden soll, an welcher Position es angezeigt werden soll und ob es persistent zu jedem Kunden mit in der Datenbank gespeichert werden soll (CRM). In dieser Klasse wird nicht nur ein Attribut gespeichert, sondern immer gleich alle existierenden Attribute (Arrays).
bestellung_def.php	Bestellung
Beschreibung **DA**	Definiert eine Klasse Bestellung, die der Forderung gerecht wird, eine ganze Bestellung inklusive all ihren Attributen und Artikel-Zuordnungen aufnehmen zu können. Die Artikel werden dabei in einem Array (für den es Zugriffsfunktionen gibt) verwaltet. Dieser Array enthält Artikel_info-Objekte (siehe weiter unten).
bestellung_def.php	Artikel_info
Beschreibung	Diese Klasse wird nur im Zusammenhang mit Bestellungen verwendet. Sie enthält Informationen bezüglich eines Artikels in einer Bestellung (also auch Anzahl u.s.w.).
kategorie_def.php	Kategorie
Beschreibung **DA**	Die Klasse Kategorie kann alle Attribute einer Kategorie fassen. In einem Array speichert das jeweilige Objekt die Unterkategorien der (Haupt-)Kategorie. Es gibt die drei Funktionen get, put und anzahl für den Unterkategorienarray.
kategorie_def.php	Unterkategorie *extends Kategorie*
Beschreibung **DA**	Unterkategorie ist eine Child-Klasse einer Kategorie. Sie übernimmt somit alle Attribute und Methoden. Die Unterkategorie erweitert die Kategorie um die Variable `$Unterkategorie_von`. Darin wird jeweils der Name der (Haupt-)Kategorie gespeichert. Diese einfache Klassenbeziehung erlaubt (eigentlich) eine beliebig tiefe Verschachtelung von Kategorien.
kreditkarte_def.php	Kreditkarte

© 2001 R. Glanzmann & J. Fontanil

Modul: Klassendefinitionen	
Beschreibung DA	Definiert die Klasse Kreditkarte. Sie speichert für jede Kreditkarte einzeln den Namen des Kreditkarten-Instituts (Hersteller), das Handling und das benutzen-Flag. Das Handling kann von der Datenbank her nur zwei Werte annehmen: 'intern', 'extern'. Mit intern ist gemeint, dass die Bestellung komplett PhPepperShop intern abgewickelt wird. Wenn 'extern' angewählt wurde, kann man ein Pay-Objekt mit der kompletten Bestellung und Kunden-Information in der Datei payment_interface.php abholen und die Bestellung als akzeptiert oder nicht_akzeptiert wieder zurück geben. Man kann die eigentliche Zahlung dann in einem externen Programm (auf einem externen Server) abwickeln.
kunde_def.php	Kunde
Beschreibung DA	Definiert die Klasse Kunde, welche alle Daten eines Kunden fassen kann. Diese Klasse wurde vor allem dafür ausgelegt, alle möglichen Daten eines Kunden speichern zu können. Wenn man nicht alle Variablen benutzt, macht das aber auch nichts. Es sollen hiermit einfach alle Möglichkeiten abgedeckt sein.
versandkosten_def.php	Versandkosten
Beschreibung DA	Diese Klasse enthält alle von der Versandkostenberechnung her benötigten Variablen und einen Array mit den Versandkonstenpreisen (z.B. ab 10kg CHF 10.00,...).
versandkosten_def.php	Versandkostenpreis
Beschreibung DA	In einer Versandkostenpreis-Klasse können alle Attribute der Tabelle versankostenpreise gespeichert werden. Man speichert pro Versandkostenpreis-Objekt ein Preisintervall (von 10kg bis 20kg kosten die Versandkosten z.B. CHF 10.00).

Modul: Datenbank Wrapper	
Datei	**Klassenname**
database.php	TRecordSet / TSybaseRecordSet / TMySQLRecordSet
Beschreibung	TRecordSet ist eine abstrakte Klasse, welche für die jeweilige Datenbank-Implementation vererbt wird. Ein RecordSet bietet Funktionen, um komfortabel und Datenbank unabhängig die einzelnen Felder auszulesen, die nächste Zeile einzulesen und die Anzahl erhaltener Zeilen auszugeben. (Funktionen: `GetField( )`, `NextRow( )`)
database.php	TDatabase / TSybaseDatabase / TMySQLDatabase
Beschreibung	TDatabase ist ebenfalls eine abstrakte Klasse, welche für die jeweilige Datenbank-Implementation weiter vererbt wird. Sie kümmert sich darum, datenbankunabängige Connections zu bieten. Dies beinhaltet ein Connection-Objekt, Zugriffsfunktionen für Queries und eine vereinfachte Zugriffsfunktion ohne Rückgabewerte (INSERT, UPDATE). Funktionen: `Exec`(SQL-Query), `Query`(SQL-Query, welche ein Resultat liefert).

Modul: Datenbankanbindung	
Datei	
initialize.php	
Beschreibung DA	Wird von User-Modulen included und stellt ein Objekt $Database zur Verfügung. Dieses Objekt eröffnet eine Verbindung als `Shopuser`. Die Verbindung ist nicht persistent und wird bei Script-Ende automatisch wieder geschlossen. Da das Login für jeden Shop wieder anders aussieht, erstellt das Installationsscript personalisierte Versionen dieser Datei aus einem Template.

 © 2001 R. Glanzmann & J. Fontanil

Modul: Datenbankanbindung	
ADMIN_initialize.php	
Beschreibung **DA**	Diese Datei wird von Administrator-Modulen eingebunden (included) und bietet eine nicht persistente Datenbank-Connection als `Shopadmin` an. Auch diese Datei wird aus einem Template erzeugt.

Modul: SQL-Statements	
Datei	
USER_SQL_BEFEHLE.php	
Beschreibung **DA**	Hier werden Variablen User-SQL-Queries zugewiesen (siehe Kapitel 2.4.7).
ADMIN_SQL_BEFEHLE.php	
Beschreibung **DA**	Hier werden wie bei USER_SQL_BEFEHLE.php die Query-Teile verschiedenen Variablen zugeordnet. Bloss handelt es sich hier um Queries von den Administrator-Funktionen. Diese Datei wird deshalb von USER-Modulen nicht included.

Modul: PHP-Funktionen - Artikel Handling	
Datei	**Funktionsname**
USER_ARTIKEL_HANDLING.php	get_Artikel
Beschreibung **DA**	Gibt auf Grund einer Artikel_ID ein Artikel-Objekt zurück. **Argumente**: Artikel_ID (INT) **Rückgabewert**: Ein Artikel (als Artikel-Objekt, siehe artikel_def.php)
USER_ARTIKEL_HANDLING.php	getArtikeleinerKategorie
Beschreibung **DA**	Alle Artikel einer Kategorie holen. Die Funktion getArtikeleinerKategorie liefert in einem Array alle Artikel, die einer Kategorie untergeordnet sind. **Argumente**: Kategoriename (String), Name der Parent-Kategorie, falls es eine Unterkategorie ist (String) **Rückgabewert**: Array (Key = Artikel_ID, Wert = Artikel-Objekt)
USER_ARTIKEL_HANDLING.php	IDgetArtikeleinerKategorie
Beschreibung **DA**	Alle Artikel einer Kategorie holen, als Argument wird eine Kategorie_ID erwartet. Die Funktion IDgetArtikeleinerKategorie liefert in einem Array alle Artikel die einer Kategorie untergeordnet sind. **Argumente**: Kategorie_ID (INT) **Rückgabewert**: Array (Key = Artikel_ID, Wert = Artikel-Objekt)
USER_ARTIKEL_HANDLING.php	getgesuchterArtikel

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen - Artikel Handling	
Beschreibung DA	Liefert aufgrund eines Such-Strings, der entweder im Namen oder in der Beschreibung des Artikels vorkommen kann, alle zutreffenden Artikel zurück. Es können jetzt auch mehrere Begriffe eingegeben werden. Diese werden dann konjunktiv verknüpft (AND). Auch Bilder werden optional angezeigt. Die beiden Argumente $low- und $highlimit definieren das Anzeigefenster: $low sagt von wo, $high sagt wieviele Artikel auf einmal ab $low ausgelesen werden sollen. **Argumente**: Artikel_ID (INT), $lowlimit (INT), $highlimit (INT) **Rückgabewert**: Array (Key: Artikel-Objekt, Wert: Kategorienarray)
USER_ARTIKEL_HANDLING.php	getallKategorien
Beschreibung DA	Liefert ein Array mit allen Kategorien als Kategorie-Objekte zurück **Argumente**: keine **Rückgabewert**: Array mit Kategorien, welche ihre Unterkategorien enthalten
USER_ARTIKEL_HANDLING.php	getKategorie_eines_Artikels
Beschreibung DA	Liefert den Kategorienamen eines gegebenen Artikels. (Wenn die Kategorie KEINE Unterkategorie ist, so ist der Parent-Kategorie-Name gleich dem Kategorienamen). **Argumente**: Artikel_ID (INT) **Rückgabewert**: Array: Key = Kategoriename (String), Wert = Parent-Kategoriename (String)
	getKategorieID_eines_Artikels
Beschreibung DA	Liefert die Kategorie_IDs der Kategorien, in welchen der angegebene Artikel eingeteilt ist. **Argumente**: Artikel_ID (INT) **Rückgabewert**: Array: Kategorie_IDs
USER_ARTIKEL_HANDLING.php	getKategorie
Beschreibung DA	Liefert als Objekt eine (Unter-)Kategorie. **Argumente**: Kategorie_ID (INT) **Rückgabewert**: Ein Unterkategorie-Objekt
USER_ARTIKEL_HANDLING.php	getvar_opt_preise
Beschreibung	Liefert die Texte und Preise aller Optionen und Variationen eines Artikels. Diese Funktion wird benötigt, da nie Preisinformationen von Seite zu Seite übertragen werden (Sicherheit!). Preisinformationen jeglicher Art, kommen immer direkt von der Datenbank in die Ziel-Page. **Argumente**: Artikel_ID (INT) **Rückgabewert**: Ein Artikel-Objekt, das Optionen und Variationen enthält

Modul: PHP-Funktionen - Userseitiges Bestellungswesen	
Datei	**Funktionsname**
USER_BESTELLUNG.php	getBestellung

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen - Userseitiges Bestellungswesen	
Beschreibung **DA**	Gibt aufgrund einer Session-ID (falls existierend) eine Bestellung als Bestellungs-Objekt zurück (inkl. allen zugeordneten Artikeln). **Argumente**: Session_ID (String) **Rückgabewert**: Bestellungs-Objekt
USER_BESTELLUNG.php	test_create_Bestellung
Beschreibung **DA**	Diese Funktion gehört einerseits zum Bestellungswesen, andererseits aber auch zum Session-Management. Sie überprüft zuerst, ob in der Datenbank eine zur aktuellen Session-ID passende Bestellung existiert. Wenn ja, ist alles gut, wenn nein, so wird eine Neue erstellt. Auf diese Weise ist sichergestellt, dass jeder Benutzer jederzeit ohne Fehlermeldung in den Bestellvorgang einsteigen kann. **Argumente**: Session_ID (String) **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	addArtikel
Beschreibung	Diese Funktion wird benötigt, wenn der Kunde auf den “In den Warenkorb“-Button klickt. Sie fügt der Bestellung des Kunden den Artikel hinzu (Zuordnung). **Argumente**: - Session_ID (String), damit man weiss welcher Bestellung der Artikel zuzuordnen ist - Artikel_info-Objekt, enthält alle für die Bestellung relevanten Felder eines Artikels (inkl. Anzahl u.s.w.) **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	del_B_Artikel
Beschreibung **DA**	Löscht einen Artikel-Eintrag für eine Bestellung. (Komische Namensgebung, weil es delArtikel schon gibt) **Argumente**: Artikel_ID (INT) und Bestellungs_ID (INT) um die betroffenen Artikel eindeutig zu bestimmen **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	delSession
Beschreibung **DA**	Löscht eine Session und damit auch die Bestellung. Es werden drei Schritte abgearbeitet: 1.) Bestellung löschen 2.) Artikelliste der Bestellung löschen 3.) Kunde-Bestellung-Referenz löschen (Tabelle bestellung_kunde) **Argumente**: - $Session_ID (String) wird benötigt um die Bestellung in der Tabelle 'bestellung' zu referenzieren. - $Bestellungs_ID (INT) wird benötigt, damit man OHNE eine weitere Query zu starten auch alle zur gelöschten Bestellung gehörenden Artikel in der artikel_bestellung-Tabelle löschen kann) **Rückgabewert**: true bei Erfolg, sonst Abbruch per die-Funktion
USER_BESTELLUNG.php	delallexpiredSessions

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen - Userseitiges Bestellungswesen	
Beschreibung DA	Löscht ALLE abgelaufenen Bestellungen und ihre Artikel- und Kunden-Referenzen. Problematik: Dies ist eine USER-Funktion, die aber eine Administrator-Operation ausführt --> Sicherheitslücke(!?). Besser wäre eine Art Cron-Job, der mit Admin-Rechten diese Operation ausführt. Wir dachten uns aber, auf diese Weise muss der Admin nichts machen und der Shop hält die Anzahl der eingetragenen Bestellungen (bei ausgeschaltetem Bestellungsmanagement) von alleine in Grenzen. Diese Funktion ist zeitaufwändig und soll an einem selten besuchten Platz aufgerufen werden (bei uns nach Bestellungs-Abschluss). **Argumente**: keine **Rückgabewert**: keiner
USER_BESTELLUNG.php	getversandkostensettings
Beschreibung DA	Liefert die Versandkosten-Einstellungen des Shops **Argumente**: Setting_Nr (im Hinblick auf mehrere zu verwaltende Shops) **Rückgabewert**: Versandkosten-Objekt
USER_BESTELLUNG.php	getversandkostentext
Beschreibung DA	Liefert den Beschreibungstext des Rechnungspostens für die Versandkosten im Warenkorb. (z.B. "Versand- und Verpackungskosten") **Argumente**: Setting_Nr (im Hinblick auf mehrere zu verwaltende Shops) **Rückgabewert**: String
USER_BESTELLUNG.php	berechneversandkosten
Beschreibung DA	Berechnet die Versandkosten einer Bestellung und schreibt die Werte auch gleich in die entsprechenden Variablen der angegebenen Bestellung. Es wird je nach Bezahlungsart noch die Nachnahmegebühr und ein Mindermengenzuschlag hinzugefügt. **Argumente**: Session_ID (String) **Rückgabewert**: Array: 1.Element = Versandkosten 2.Element = Mindermengenzuschlag 3.Element = Rechnungstotal (ALLES inkl. allem) 4.Element = Nachnahmegebühr
USER_BESTELLUNG.php	getKunde
Beschreibung DA	Gibt aufgrund einer Kunden_ID ein Kunden-Objekt zurück. **Argumente**: Kunden_ID (String) **Rückgabewert**: Ein Kunde in Form eines Kunden-Objekts (siehe kunde_def.php)
USER_BESTELLUNG.php	getallKunden
Beschreibung DA	Gibt in einem Array von Kunden-Objekten alle Kunden zurück. **Argumente**: keine **Rückgabewert**: Array mit Kunden-Objekten (siehe kunde_def.php)
USER_BESTELLUNG.php	newKunde

© 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen - Userseitiges Bestellungswesen	
Beschreibung DA	Fügt der Datenbank einen neuen Kunden hinzu (betroffene Tabellen kunde, bestellung_kunde). Dabei wird auch eine neue Kunden_ID berechnet, siehe Kapitel Security, Kunden-ID Kapitel 2.1.6. **Argumente**: Ein Kunde in 'Einzelteilen': (vom Formular her) $Kunden_ID,$Session_ID,$Kunden_Nr,$Anrede,$Vorname,$Nachname,$Firma,$Abteilung,$Strasse,$Postfach,$PLZ,$Ort,$Land,$Tel,$Fax,$Email,$Einkaufsvolumen,$Login,$Passwort,$gesperrt,$temp,$Attribut1,$Attribut2,$Attribut3,$Attribut4,$Attributwert1,$Attributwert2,$Attributwert3,$Attributwert4 **Rückgabewert**: Die Kunden_ID (String) des neuen Kunden
USER_BESTELLUNG.php	delKunde
Beschreibung DA	Löscht einen Kunden in der Datenbank. **Argumente**: Kunden_ID (String) **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	getKunde_einer_Bestellung
Beschreibung DA	Gibt aufgrund einer Bestellungs_ID ein Kunden-Objekt zurück. **Argumente**: Bestellungs_ID (INT) **Rückgabewert**: Kunden_Objekt (siehe kunde_def.php)
USER_BESTELLUNG.php	updKunde
Beschreibung DA	Dies ist ein Update aller Kundendaten. (Diese Funktion wird momentan nicht verwendet, erlaubt aber das Setzen ALLER Kunden-Attribute) **Argumente**: Kunde in Einzel-Attributen: (von einem Formular her) $Kunden_Nr,$Session_ID,$Anrede,$Vorname,$Nachname,$Firma,$Abteilung,$Strasse,$Postfach,$PLZ,$Ort,$Land,$Tel,$Fax,$Email,$Einkaufsvolumen,$Login,$Passwort,$gesperrt,$temp,$Attribut1,$Attribut2,$Attribut3,$Attribut4,$Attributwert1,$Attributwert2,$Attributwert3,$Attributwert4,$Kunden_ID **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	updKundenFelder
Beschreibung DA	Dies ist ein Update aller vom Kunden beinflussbaren / benötigten Felder seines Datensatzes. **Argumente**: Kundendaten in Einzelteilen, vom Formular her: $Session_ID,$Anrede,$Vorname,$Nachname,$Firma,$Abteilung,$Strasse,$Postfach,$PLZ,$Ort,$Land,$Tel,$Fax,$Email,$Attribut1,$Attribut2,$Attribut3,$Attribut4,$Attributwert1,$Attributwert2,$Attributwert3,$Attributwert4,$Kunden_ID **Rückgabewert**: true bei Erfolg, sonst Abbruch per die-Funktion
USER_BESTELLUNG.php	gibBestellung_an_Kunde
Beschreibung DA	Ordnet eine Bestellung einem Kunden zu (in Tabelle bestellung_kunde). **Argumente**: Bestellungs_ID (INT), Kunden_ID (String) **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	delBestellung_von_Kunde

© 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen - Userseitiges Bestellungswesen	
Beschreibung DA	Löscht alle Bestellungen eines Kunden. Wird benötigt um persistente Kunden, ihre Bestellungen und deren Referenzen zum zu löschenden Kunden zu entfernen. Zuerst werden alle zu löschenden Bestellungs_IDs ausgelesen, dann alle Referenzen vom Kunden zu den Bestellungen gelöscht und zum Schluss die Bestellungen entfernt. **Argumente**: Kunden_ID (String) **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	getAttributobjekt
Beschreibung DA	Gibt ein Attribut-Objekt zurück (darin befinden sich in Arrays alle Attribute, sortiert nach ihrer jeweiligen Positions-Nr). **Argumente**: keine **Rückgabewert**: Attribut-Objekt
USER_BESTELLUNG.php	setAttributobjekt
Beschreibung DA	Setzt alle Attribute in der Tabelle attribut. **Argumente**: Attribut-Objekt **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	checkLogin
Beschreibung DA	Diese Funktion überprüft die eingegebenen Login-Daten und veranlasst dann, je nach Eingabe: - einen neuen temporären Kunden anzulegen - die Kunden_ID eines persistenten Kunden auszulesen und seine Session_ID upzudaten **Argumente**: Login (String), Passwort (String), Session_ID (String) **Rückgabewert**: - User existiert, Passwort stimmt - -> existierende Kunden_ID (String) - User existiert, Passwort stimmt nicht - -> "P" (String) - User existiert nicht - -> neue Kunden_ID (String)
USER_BESTELLUNG.php	checkSession
Beschreibung DA	Diese Funktion gibt die Kunden_ID zurück, falls ein Kunde eingeloggt ist und sich schon authentifiziert hat. **Argumente**: Session_ID (String) **Rückgabewert**: User existiert und ist eingeloggt --> Kunden_ID (String) User ist nicht eingeloggt --> "" (Leer-String)
USER_BESTELLUNG.php	mailPasswort

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen - Userseitiges Bestellungswesen	
Beschreibung DA	Diese Funktion sendet dem Benutzer, der sein Passwort vergessen hat, sein Passwort an seine im System gespeicherte E-Mail Adresse. Zur 'Verifikation' muss er sein Login eingeben. (Zuordnung zur E-Mail-Adresse). **Argumente**: Login (String) **Rückgabewert**: true bei Erfolg false wenn kein Login gefunden wurde sonst Abbruch per die-Funktion
USER_BESTELLUNG.php	addEinkaufsvolumen
Beschreibung DA	Diese Funktion addiert das Bestellungstotal zum Einkaufsvolumen eines Kunden, ebenso wird das LetzteBestellung-Attribut erneuert. **Argumente**: Session_ID (String), Betrag (Double) **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	updBestellungsFelder
Beschreibung DA	Hier werden die zur Bestellung gehörenden Daten eines Kunden zwischengespeichert (Anmerkung, Datum, Bezahlungsart, Kreditkartendaten, Zusatzattribute1 bis 4 und ihre Namen) **Argumente**: Session_ID (String) (Damit man weiss, zu welcher Bestellung die Daten gehören werden. Bestellungs-Objekt enthält die User-Daten (Def. siehe bestellung_def.php)) **Rückgabewert**: true bei Erfolg
USER_BESTELLUNG.php	getEmailMessage
Beschreibung DA	Da bei der Kreditkartenzahlung mit externer Zahlungsabwicklung der E-Mail-Message String verloren gehen würde, wird dieser temporär in der jeweiligen Bestellung zwischengespeichert. Diese Funktion liest den temporär zwischengespeicherten E-Mail-Message-String aus und LÖSCHT den E-Mail-Message-String in der entsprechenden Bestellung! **Argumente**: Session_ID (String) **Rückgabewert**: String oder Abbruch per die-Funktion
USER_BESTELLUNG.php	putEmailMessage
Beschreibung DA	Da bei der Kreditkartenzahlung mit externer Zahlungsabwicklung der E-Mail-Message String verloren gehen würde, wird dieser temporär in der jeweiligen Bestellung zwischengespeichert. Mit dieser Funktion kann man den E-Mail-Message-String temporär der Bestellung des Kunden zuweisen. **Argumente**: Emailstring (String), Session_ID (String) **Rückgabewert**: true bei Erfolg oder Abbruch per die-Funktion
USER_BESTELLUNG.php	validateCC
Beschreibung DA	Die folgende Funktion *von Error Brett (Brett@InterWebDesign.com)* überprüft die Kreditkartennummer auf Vollständigkeit und Prüfsumme (MOD-10 Verfahren). Das Datum wird nicht hier validiert. **Argumente**: Kreditkartennummer (String), Kreditkarten_Institut (String) **Rückgabewert**: - true bei korrekter Kreditkartennummer - false bei falscher Kreditkartennummer - (-1) bei nicht korrektem Kreditkarten Institut
USER_BESTELLUNG.php	toCharArray

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen - Userseitiges Bestellungswesen	
Beschreibung DA	Diese Funktion gibt eine Zeichenkette als Array zurück. Sie wird bis jetzt nur von validateCC benötigt. **Argumente**: String **Rückgabewert**: Array der Zeichen vom Eingabestring
USER_BESTELLUNG.php	schliessenBestellung
Beschreibung DA	Damit ein Kunde seine Bestellung nicht mehr abändern kann, wird seine Bestellung mit dem Attribut Bestellung_abgeschlossen in der Tabelle bestellung als unveränderbar markiert. In der Datenbank wird ferner die Session-ID bei diesem Eintrag gelöscht. Somit erhält er mit der gleichen Session-ID eine neue Bestellung in der Datenbank (siehe auch test_create_Session). **Argumente**: Session_ID (String), abschliessen_Flag enum('Y','N') **Rückgabewert**: true bei Erfolg, sonst Abbruch per die-Funktion

Modul: PHP-Funktionen – Einzelne Shop-Settings auslesen	
Datei	**Funktionsname**
USER_ARTIKEL_HANDLING.php	getWaehrung
Beschreibung	Liefert die Währung in einem String zurück. (z.B. SFr., CHF, DM, ATS) **Argumente**: keine **Rückgabewert**: Währung (String, max. 4 Zeichen)
USER_ARTIKEL_HANDLING.php	getGewichts_Masseinheit
Beschreibung	Liefert die Gewichts-Masseinheit in einem String zurück. (z.B. kg, ml, g, t,...) **Argumente**: keine **Rückgabewert**: Gewichts-Masseinheit (String)
USER_ARTIKEL_HANDLING.php	getThumbnail_Breite
Beschreibung	Da in bild_up.php ein Thumbnail eines Artikel-Bildes erzeugt wird, kann über dieses shop_settings-Attribut die Breite des Thumbnails festgelegt werden. Diese Funktion liefert nun den Integer-Wert in Pixel. **Argumente**: keine **Rückgabewert**: Thumbnail-Breite (INT)
USER_ARTIKEL_HANDLING.php	getshopname
Beschreibung	Liefert den in den Shop-Settings gespeicherten Namen des Shops in einem String. **Argumente**: keine **Rückgabewert**: Shopname (String)
USER_ARTIKEL_HANDLING.php	getmax_session_time
Beschreibung	In den Shop-Settings lässt sich ein fixer Wert in Sekunden einstellen, der als die maximal gültige Dauer einer Session zählt. So ist man etwas freier bei der Wahl der Session-Dauer. (Nach oben gibt es die Limite in der `php.ini`). Diese Funktion liefert nun diesen Wert als Integer zurück. **Argumente**: keine **Rückgabewert**: Maximale Dauer einer User-Session (INT)

© 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Einzelne Shop-Settings auslesen	
USER_ARTIKEL_HANDLING.php	getBezahlungsart
Beschreibung	Liefert einen Array, in dem steht, welche Zahlungsarten gewählt werden können. Der jeweilige Name ist 'Y', wenn er in den Shop-Settings aktiviert wurde, sonst 'N'. **Argumente**: keine **Rückgabewert**: Array (Key = Zahlungsart, Wert = 'Y' oder 'N')
USER_ARTIKEL_HANDLING.php	getAGB
Beschreibung	Liefert die Allgemeinen Geschäftsbedingungen (AGB) aus der Datenbank als String (Tabelle `shop_settings`). **Argumente**: keine **Rückgabewert**: Allgemeine Geschäftsbedingungen (String)
USER_ARTIKEL_HANDLING.php	getEmail
Beschreibung	Liefert die E-Mail Adresse als String passend zur Bestellung, welche von der im Argument angegebenen Session_ID referenziert wird (Kunden E-Mail Adresse). **Argumente**: Session_ID (String) **Rückgabewert**: E-Mail-Adresse als String (Eingaben ungeprüft!)
USER_ARTIKEL_HANDLING.php	getSuchInkrement
Beschreibung **DA**	Liefert das Such-Inkrement als Integer --> Wieviele gefundene Resultate auf einmal anzeigen (siehe Artikelsuche) **Argumente**: keine **Rückgabewert**: Such-Inkrement (INT)
USER_ARTIKEL_HANDLING.php	getBestellungsmanagement
Beschreibung **DA**	Liefert 'Y' oder 'N', je nachdem ob das Bestellungsmanagement erwünscht ist oder nicht. Eingestellt wird dieses Attribut in den allgemeinen Shop Einstellungen (setshopsettings). **Argumente**: keine **Rückgabewert**: 'Y' oder 'N' (String)
USER_ARTIKEL_HANDLING.php	getvaroptinc
Beschreibung **DA**	Liest aus den Shop-Settings aus, wieviele Felder für die Erfassung von Optionen und Variationen bei einem Artikel mindestens dargestellt werden. Zusätzlich wird noch der Wert ausgelesen, wie viele leere Felder angezeigt werden, wenn die Anzahl der eingegeben Optionen / Variationen die Anzahl der eingestellten Minimalfelderanzahl übersteigt. **Argumente**: keine **Rückgabewert**: *entweder* Array: 1.Element = Optionen Inkrement (Opt_inc), 2. Element = Variations Inkrement (Var_inc) 3. Element = Optionen Basisanzahl (Opt_anz) 4. Element = Variations Basisanzahl (Var_anz) *oder* Abbruch per die-Funktion

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Einzelne Shop-Settings auslesen	
USER_ARTIKEL_HANDLING.php	getSSL
Beschreibung DA	Diese Funktion überprüft, ob SSL eingeschaltet wurde. Wenn ja, wird eine absolute URL zusammengebaut und `https` vorangestellt. Um wieder aus dem SSL-Bereich austreten zu können, kann man der Funktion im Flag $noSSL (true) mitteilen, dass man den SSL-Bereich verlassen will. Es wird dann als Rückgabewert eine URL mit `http` zurückgegeben. Mit der Variable $server wird meistens $SERVER_NAME übergeben. Ein Aufruf dieser Funktion sieht typischerweise so aus: getSSL($PHP_SELF, $SERVER_NAME, false). Eingestellt wird SSL in den allgemeinen Shop Einstellungen (setshopsettings) **Argumente**: $oldphpself (String, $PHP_SELF), $server (String, $SERVER_NAME), $noSSL (boolean) **Rückgabewert**: Neuer URL (mit oder ohne https) (String)
USER_ARTIKEL_HANDLING.php	getNachnahmebetrag
Beschreibung DA	Liefert den Geldwert, welcher bei einer Nachnahmelieferung zu verrechnen ist (Nachnahmegebühr). **Argumente**: keine **Rückgabewert**: Nachnahmebetrag (DOUBLE), 0.0 = keiner vorhanden
USER_ARTIKEL_HANDLING.php	getKontoinformation
Beschreibung DA	Liefert den String, welchen man in den allgemeinen Shop-Einstellungen neben Vorauskasse im Feld Kontoinformation eingeben kann. **Argumente**: keine **Rückgabewert**: Kontoinformation (String)
USER_ARTIKEL_HANDLING.php	getcssarg
Beschreibung DA	Einen CSS-String aus der Tabelle css_file auslesen: **Argumente**: CSS-ID **Rückgabewert**: zugehöriger CSS-String
USER_ARTIKEL_HANDLING.php	getKreditkarten
Beschreibung DA	Diese Funktion liest alle Kreditkartendaten aus der kreditkarten-Tabelle aus und gibt alle Kreditkarten-Objekte in einem Array zurück. **Argumente**: keine **Rückgabewert**: Array von Kreditkarten-Objekten
USER_ARTIKEL_HANDLING.php	getShopEmail
Beschreibung	Liefert die E-Mail Adresse des Shop Betreibers (oder des Verantwortlichen - -> in Shop-Settings eingestellt) **Argumente**: keine **Rückgabewert**: Shop-Email-Adresse (String)

Modul: PHP-Funktionen – Administrationsfunktionen	
Datei	**Funktionsname**
SHOP_ADMINISTRATION.php	newArtikel

© 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Administrationsfunktionen	
Beschreibung **DA**	Einen neuen Artikel in die Datenbank speichern. Diese Funktion fügt einen neuen Artikel ohne Bild in die Datenbank ein. Sie wird von der Datei bild_up.php aufgerufen. (Betrifft Tabellen artikel, artikel_optionen, artikel_variationen, artikel_kategorien). Damit mehrere Kategorien pro Artikel möglich sind, werden die Kategorie_IDs in einem Array übergeben. Eine weitere Spezialität ist das Flag $kat_save. Nur wenn dieses = true (1) ist, wird der neue Artikel auch in die angegebenen Kategorien eingetragen (siehe auch updArtikel_2) **Argumente**: In Einzelteile zerlegter Artikel, vom Formular her: $Kategorie_IDarray,$Artikel_Nr,$Name,$Beschreibung,$letzteAenderung,$Preis,$Aktionspreis,$Gewicht,$Link,$Optionenarray,$Variationsarray,$kat_save **Rückgabewert**: Artikel_ID (INT) des neu eingefügten Artikels. (Artikel_ID wurde per Auto-Increment Funktion erzeugt (MySQL). Hier sei noch erwähnt, dass der DB-Wrapper (database.php) nur für MySQL diese Funktionalität bietet. Bei den anderen Datenbanken, auch Sybase, muss das noch nachprogrammiert werden.
SHOP_ADMINISTRATION.php	delArtikel
Beschreibung **DA**	Einen Artikel aus der Shop-Datenbank entfernen (Betrifft mehrere Tabellen). **Argumente**: Artikel_ID (INT) **Rückgabewert**: 1 (entspricht true) bei Erfolg
SHOP_ADMINISTRATION.php	updArtikel
Beschreibung	Einen schon bestehenden Artikel in der Datenbank verändern (updaten): Diese Funktion ist etwas verzettelt. Sie wurde seit der Version 1.05 komplett überarbeitet und zum grossen Teil neu programmiert. Auf die Schnelle erklärt, wird ein Artikel wie folgt upgedated: 1.) Daten des zu updatenden Artikels einlesen 2.) Per darstellenArtikel-Funktion (SHOP_ADMINISTRATION_ARTIKEL.php) diesen im Eingabeformular darstellen [3.) Optionaler Bild-Einfüge-Dialog] (bild_up.php, hier drin geht es auch weiter mit dem Update) 4.) Über die updArtikel_2 Funktion den Artikel updaten (siehe gleich unter dieser Funktion) **Argumente**: Artikel_ID (INT) **Rückgabewert**: - Artikel_ID bei Erfolg - false, bei erfolgreichem Artikel Update- der Artikel wurde in die Kategorie Nichtzugeordnet abgelegt - Abbruch der Funktion per die-Funktion und entsprechender Fehlermeldung
SHOP_ADMINISTRATION.php	updArtikel_2

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Administrationsfunktionen	
Beschreibung **DA**	Diese Funktion wird von der Datei bild_up.php aufgerufen, sobald ein Artikel upgedated werden soll. Dies ist quasi der zweite Teil eines Artikel Updates. Diese Funktion updated nur die Artikeldaten OHNE das Bild (Weiteres zum eigentlichen Bild, siehe bild_up.php). Das Flag $kat_save wird benutzt um den Artikel upzudaten, aber nicht um die Kategorien-Referenzen des Artikels zu aktualisieren (MEHR-OPTIONEN / VARIATIONEN Button Implementation). Es bewirkt, dass Teil 4 von updArtikel_2 übersprungen wird. **Argumente**: "In Einzelteile zerlegter Artikel, vom Formular her" $Kategorie_IDarray,$Artikel_ID,$Artikel_Nr,$Name,$Beschreibung,$letzteAenderung,$Preis,$Aktionspreis,$Gewicht,$Link,$Optionenarray,$Variationsarray,$kat_save **Rückgabewert**: Artikel_ID (INT) bei Erfolg
SHOP_ADMINISTRATION.php	movengzArtikel
Beschreibung **DA**	Kopiert den angegebenen Artikel in die Kategorie 'Nichtzugeordnet'. **Argumente**: Artikel_ID (INT) **Rückgabewert**: true bei Erfolg
SHOP_ADMINISTRATION.php	getshopsettings
Beschreibung **DA**	Liefert die in der Datenbank in der Tabelle shop_settings definierten Shop-Einstellungen als einen Array of String. **Argumente**: keine **Rückgabewert**: Array: Keys = Feldnamen, Werte = Werte der Felder
SHOP_ADMINISTRATION.php	setshopsettings
Beschreibung **DA**	Macht ein Update auf alle Attribute der Tabelle shop_settings und aktualisiert somit alle Shop-Settings. **Argumente**: Alle Shopsettings einzeln (dies ist praktisch, da diese Funktion nur nach einem Formular aufgerufen wird): $MwStsatz,$MwStpflichtig,$MwStNummer,$Admin_pwd,$Name,$Adresse1,$Adresse2,$PLZOrt,$Tel1,$Tel2,$Email,$Thumbnail_Breite,$Mindermengenzuschlag_Aufpreis,$Abrechnung_nach_Preis,$Abrechnung_nach_Gewicht,$Mindermengenzuschlag,$Kreditkarten_Postcard,$ShopVersion,$Abrechnung_nach_Pauschale,$Rechnung,$ShopVersion,$Waehrung,$Nachnahme,$Mindermengenzuschlag_bis_Preis,$keineVersandkostenmehr_ab,$keineVersandkostenmehr,$SSL,$Bestellungsmanagement,$Gewichts_Masseinheit,$max_session_time,$AGB,$Opt_inc,$Var_inc,$Opt_anz,$Var_anz,$SuchInkrement,$Vorauskasse,$Kontoinformation **Rückgabewert**: true bei Erfolg
SHOP_ADMINISTRATION.php	delBild
Beschreibung	Löscht das Bild des aktuellen Artikels. Löscht das Bild eines Artikels in der DB (Wenn man dem Artikel kein Bild mehr zuordnen will. Diese Funktion wird beim Update eines Artikels verwendet). **Argumente**: Artikel_ID (INT) **Rückgabewert**: true bei Erfolg
SHOP_ADMINISTRATION.php	newKategorie

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Administrationsfunktionen	
Beschreibung **DA**	Fügt eine neue Kategorie in die Tabelle 'kategorien' ein. Unterstützt vorerst NUR den Namen und die Positionsnummer. Es ist noch keine Beschreibung und Bild-Eingabe möglich. Seit v.1.05 werden auch Unterkategorien unterstützt. **Argumente**: Name (String), Positions-Nr (INT), Unterkategorie_von (String) der neuen Kategorie / Unterkategorie. **Rückgabewert**: true bei Erfolg
SHOP_ADMINISTRATION.php	delKategorie
Beschreibung **DA**	Eine Kategorie löschen: Dabei werden die noch darin enthaltenen Artikel, sofern sie keiner weiteren, noch existiernden Kategorie angehören, in die Kategorie 'Nichtzugeordnet' abgelegt. Diese Funktion benötigt seit der Implementierung von Unterkategorien und Mehrfachkategorien pro Artikel eine komplexe Logik. Ist das Artikelloeschen_Flag gleich true, so werden die Artikel gelöscht. **Argumente**: Kategorie_ID (INT), Artikelloeschen_Flag (boolean) **Rückgabewert**: true bei Erfolg
SHOP_ADMINISTRATION.php	verschiebenKategorie
Beschreibung **DA**	Eine (Unter-)Kategorie verschieben. Die Funktionalität, um die Positionsnummern nach dem Verschieben wieder richtig zu verteilen, wurde in die Funktion katposschieben verlegt. **Argumente**: Kategorie_ID (INT), neue Positionsnummer (INT), aktuelle- und neue Unterkategorie (Strings) **Rückgabewert**: true bei Erfolg
SHOP_ADMINISTRATION.php	umbenennenKategorie
Beschreibung **DA**	Eine (Unter-)Kategorie umbenennen. Es wird der Name der entsprechenden Kategorie aktualisiert. Falls die Kategorie Unterkategorien besitzt, wird deren Attribut Unterkategorie_von nachgeführt. **Argumente**: Kategorie_ID (INT), Neuer_Name (String) **Rückgabewert**: true oder Abbruch per die-Funktion
SHOP_ADMINISTRATION.php	katposschieben
Beschreibung **DA**	Diese Funktion ist eine Hilfsfunktion im Kategorien-Management. Sie verschiebt die Positions-Nummern der Kategorien entsprechend den Eingabe-Parametern. Positions-Nummern ordnen die Kategorien-Anzeige aufsteigend ein. Nachdem diese Funktion abgelaufen ist, kann man (z.B.) entweder eine neue (Unter-)Kategorie mit gewünschter Pos_Nr eintragen (eine Lücke wurde geschaffen), oder es wurde eine Lücke gelöscht --> z.B. wie bei delKategorie. **Argumente**: Kategorie_ID (INT), $currentPos (INT), $newPos (INT), $delPos (INT), $currentUkat (String), $newUkat (String) **Rückgabewert**: true oder Abbruch per die-Funktion
SHOP_ADMINISTRATION.php	getNichtzugeordnetKategorie

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Administrationsfunktionen	
Beschreibung DA	Liefert als Kategorie-Objekt die spezielle Nichtzugeordnet Kategorie. Diese Kategorie wird erkannt, indem sie in ihrem Attribut Unterkategorie_von den String *@PhPepperShop@* trägt. Diese Funktion wurde geschrieben, weil die Kategorie in einem englischen Shop wahrscheinlich nicht mehr 'Nichtzugeordnet' heissen wird. Auf diese Weise kann man diese Kategorie immer auslesen, egal wie sie heisst. **Argumente**: keine **Rückgabewert**: ein Kategorie-Objekt
SHOP_ADMINISTRATION.php	updatecssarg
Beschreibung DA	Einen CSS-String in der Tabelle css_file updaten. **Argumente**: CSS-ID (INT), CSS-String (String) **Rückgabewert**: true, sonst Funktionsabbruch per die
SHOP_ADMINISTRATION.php	setvaroptinc
Beschreibung DA	Die Inkremente für weitere leere Felder in der darstellen_Artikel-Funktion setzen (Tabelle shop_settings, Attribute: Opt_inc, Var_inc). **Argumente**: Optionsinkrement (INT), Variationsinkrement (INT) **Rückgabewert**: Entweder true (1) oder Abbruch per die-Funktion
SHOP_ADMINISTRATION.php	setversandkostensettings
Beschreibung DA	Speichert alle Versandkosten-Einstellungen in der Datenbank ab. **Argumente**: Man gibt alle Versandkostensettings einzeln an. Dies ist praktisch, da diese Funktion nur nach einem Formular aufgerufen wird: $Abrechnung_nach_Preis,$Abrechnung_nach_Gewicht,$Abrechnung_nach_Pauschale,$Pauschale_text,$keineVersandkostenmehr,$keineVersandkostenmehr_ab,$Mindermengenzuschlag,$Mindermengenzuschlag_bis_Preis,$Mindermengenzuschlag_Aufpreis,$Nachnamebetrag,$Setting_Nr,$Versandkostenpreise **Rückgabewert**: Die Funktion liefert true bei Erfolg
SHOP_ADMINISTRATION.php	getBestellung_Ref
Beschreibung DA	Diese Funktion liefert eine Bestellung aufgrund einer Referenz-Nr zurück. Eine Referenz_Nr ist die Summe aus dem Offset 154870 und der aktuellen Bestellungs_ID. Die Referenznummer wird dem Kunden auf seinem E-Mail angegeben und soll z.B. bei Anfragen das Suchen nach Besetellungen vereinfachen. **Argument**: Referenz_Nr (INT) **Rückgabewert**: Eine Bestellung als Bestellungs-Objekt
SHOP_ADMINISTRATION.php	getBestellung_Kunde
Beschreibung DA	Gibt auf Grund eines Kundennamens die dazugehörende(n) Bestellung(en) zurück. **Argument**: Name und Vorname des Kunden (Strings) **Rückgabewert**: Ein Array von Bestellung(-en) als Bestellungs-Objekt(-e) (Definition siehe bestellung_def.php)
SHOP_ADMINISTRATION.php	getBestellung_Alle

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Administrationsfunktionen	
Beschreibung DA	Gibt alle als abgeschlossen markierten Bestellungen in einem Array zurück. **Argumente**: keine **Rückgabewert**: Ein Array von Bestellungen als Bestellungs-Objekte
SHOP_ADMINISTRATION.php	delBestellung
Beschreibung DA	Diese Funktion löscht eine abgeschlossene Bestellung unwiderruflich. Natürlich werden auch all ihre Referenzen in der Tabelle artikel_bestellung gelöscht. Als dritter Teil werden noch alle Referenzen zum Kunden der Bestellung gelöscht. **Argumente**: Bestellungs_ID (INT) **Rückgabewert**: true bei Erfolg
SHOP_ADMINISTRATION.php	setKreditkarten
Beschreibung DA	Diese Funktion schreibt alle Kreditkartendaten in die kreditkarten-Tabelle (Speichern der Kreditkarten-Institute und ihrer Optionen). Vorgehen: Vorandene Daten löschen, neue Daten speichern (Kategorie_ID inkrementiert) **Argumente**: Herstellerarray (Array of String), benutzenarray (Array of enum('Y','N')), Handlingarray (Array of enum('Y','N')) **Rückgabewert**: true bei Erfolg

Modul: PHP-Funktionen – Shop Layout	
Datei	**Funktionsname**
SHOP_LAYOUT.php	mkindexphp
Beschreibung DA	Erstellt die Datei index.php aus dem Template-File indextemplate.txt. Tags, welche im Templatefile in Doppelklammern (<< >>) stehen, werden durch die entsprechenden Einträge in der Datenbank (Tabelle: cssfile) ersetzt. **Argumente**: keine **Rückgabewert**: false, wenn ein Fehler aufgetreten ist
SHOP_LAYOUT.php	mkcssfiles
Beschreibung DA	Erstellt die CSS-Dateien (shopstyles.css) in den Verzeichnissen shop, shop/Frameset und shop/Admin aus dem Templatefile csstemplate.txt. Tags, welche im Templatefile in Doppelklammern (<< >>) stehen, werden durch die entsprechenden Einträge in der Datenbank (Tabelle: cssfile) ersetzt. **Argumente**: keine **Rückgabewert**: false, wenn ein Fehler aufgetreten ist
SHOP_LAYOUT.php	htmltext

 © 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Shop Layout	
Beschreibung **DA**	Gibt eine Tabelle im HTML-Format aus, die folgende Elemente enthält: - R,G,B Editierfelder für Schriftfarbe - dropdown-Box für Schriftgewicht - dropdown-Box für Schriftgrösse - dropdown-Box für Text Decoration Die Parameter werden aus der Datenbank ausgelesen und direkt in die angezeigten Felder eingefüllt **Argumente**: $id -> identifier in datenbank $label -> Benennung des Tags, z.B ShopTitel $color -> wenn 1 (true) wird Farbauswahl angezeigt $weight -> wenn 1 (true) wird Schriftgewicht angezeigt $size -> wenn 1 (true) wird Schriftgrösse angezeigt $style -> wenn 1 (true) wird Text Style angezeigt **Rückgabewert**: HTML Tabelle mit oben genannten Formularelementen. Formular Bezeichner bestehen aus der Variablen $id und folgendem Text.: _c -> für RGB-Wert (color) _w -> für Schriftgewicht (width) _s -> für Schriftgrösse (size) _d -> für Text-Dekoration (decoration) _i -> für Text-Stil (style)
SHOP_LAYOUT.php	savefont
Beschreibung **DA**	Speichert die Fonteinstellungen eines Font-Tags in der DB. Das Tag wird um folgende Erweiterungen ergänzt in der CSS-Tabelle gespeichert: _c -> für RGB-Wert (color) _w -> für Schriftgewicht (width) _s -> für Schriftgrösse (size) _d -> für Text-Dekoration (decoration) **Argumente**: Font-tags: $font_tag, $c_wert, $w_wert, $s_wert, $d_wert, $i_wert **Rückgabewert**: true bei Erfolg
SHOP_LAYOUT.php	rgbshow
Beschreibung **DA**	Holt einen RGB-CSS-Wert aus der DB und gibt ihn als Teil eines HTML-Formulars mit Eingabefeldern für R,G,B aus. **Argumente**: rgb-identifier **Rückgabewert**: HTML-Formularteil mit Eingabemöglichkeit für R,G,B
SHOP_LAYOUT.php	getnocomma
Beschreibung **DA**	Holt einen CSS-Wert aus der DB und entfernt allfällig vorhandene Kommas. Wird für das Fontset benötigt. **Argumente**: CSS-Identifier **Rückgabewert**: CSS-String ohne Kommas

© 2001 R. Glanzmann & J. Fontanil

Modul: PHP-Funktionen – Shop Layout	
SHOP_LAYOUT.php	rgbdechex
Beschreibung **DA**	Wandelt die dezimal in einem Array übergebenen Farbkomponenten (R,G,B) in den für das CSS-File notwendigen Hex-String um. **Argumente**: Farbkomponenten-Array [0] = R {0..255} [1] = G {0..255} [2] = B {0..255} **Rückgabewert**: RGB-Hex-String im Format (#RRGGBB)
SHOP_LAYOUT.php	rgbhexdec
Beschreibung **DA**	Wandelt einen Hex-String in der Form #RRGGBB in die entsprechenden Farbkomponenten (R,G,B) im Dezimalformat um und gibt diese in einem Array zurück: **Argumente**: HTML-Hex-String Format: # RRGGBB **Rückgabewert**: Array mit den Farbkomponenten R,G,B im Dezimalformat [0] = R [1] = G [2] = B

 © 2001 R. Glanzmann & J. Fontanil

Modul: Aufruf-Module – Artikel Handling	
Datei	**Variable $darstellen =**
USER_ARTIKEL_ HANDLING_AUFRUF.php	1
Beschreibung **DA**	Alle Artikel einer Kategorie im Content-Frame darstellen. Wird das Artikel-Thumbnail angeklickt, geht ein Pop-Up Fenster auf, worin das Artikelbild in Originalgrösse angezeigt wird.
USER_ARTIKEL_ HANDLING_AUFRUF.php	30
Beschreibung **DA**	Artikel Suchen 1/2. Eingabefenster einblenden (Eingabe von einem oder mehreren Suchstrings, Bilder anzeigen Ja/Nein), danach weiter im darstellen = 3.
USER_ARTIKEL_ HANDLING_AUFRUF.php	3
Beschreibung **DA**	Artikel Suchen 2/2: Verarbeitung der Eingabe und Ausgabe des formatierten Resultats
USER_ARTIKEL_ HANDLING_AUFRUF.php	4
Beschreibung **DA**	Im left-Frame alle Kategorien zur Auswahl anzeigen (Auswahlmenü). Die selektierte Kategorie oder Unterkategorie wird in der Variable $selected übergeben. Darin steht die Kategorie_ID der entsprechenden Kategorie, resp. Unterkategorie. (Diese Funktion wird von ../index.php aufgerufen -> siehe dortiges Frameset)

Modul: Aufruf-Module – Userseitiges Bestellungswesen Teil 1/2	
Datei	**Variable $darstellen =**
USER_BESTELLUNG_AUFRUF.php	1
Beschreibung **DA**	Anzeige des Warenkorbs des Kunden im Content-Frame.
USER_BESTELLUNG_AUFRUF.php	2
Beschreibung	Wird aufgerufen, sobald der Kunde einen Artikel in seinen Warenkorb gelegt hat. (Weiche: Zurück zu Artikel, in Warenkorb- oder zur Kasse gehen)
USER_BESTELLUNG_AUFRUF.php	3
Beschreibung	Wenn ein Kunde in seinem Warenkorb bei einem Artikel auf löschen klickt, wird dieser Meldungsbildschirm angezeigt

Modul: Aufruf-Module – Userseitiges Bestellungswesen Teil 2/2	
Datei	**Variable $darstellen =**
USER_BESTELLUNG_1.php	1

 © 2001 R. Glanzmann & J. Fontanil

Modul: Aufruf-Module – Userseitiges Bestellungswesen Teil 2/2	
Beschreibung **DA**	Anzeige des Warenkorbs des Kunden im Content-Frame. Bevor der Login-Screen ausgegeben wird, prüfen ob sich der Kunde schon eingeloggt hat: ja --> direkt darstellen = 11 aufrufen nein --> Login-Screen ausgeben (darstellen=1) Login-Screen für Kunde, falls nicht zu darstellen = 11 gesprungen wurde.
USER_BESTELLUNG_1.php	9
Beschreibung **DA**	Eingabeformular, falls jemand sein Passwort vergessen hat.
USER_BESTELLUNG_1.php	10
Beschreibung **DA**	Passwort einem User zuschicken
USER_BESTELLUNG_1.php	11
Beschreibung **DA**	Eingabeformular für Lieferadresse und Zahlungsart
USER_BESTELLUNG_1.php	2
Beschreibung **DA**	- Update der Kunden- und Bestellungsdaten. - Ausgabe der Bestellung zur Kontrolle für den Kunden (an den Browser). - Weiche für Bezahlungsarten. Je nachdem ob man entweder per Vorauskasse, Rechnung, Nachnahme oder per Kreditkarte bezahlt und weiter noch, je nachdem ob die Kreditkartenzahlung intern oder extern abgewickelt wird.
USER_BESTELLUNG_1.php	3
Beschreibung **DA**	AGB Weiche bei Vorauskasse-, Rechnung- und Nachnahme-Zahlungen: Allgemeine Geschäftsbedingungen aus Datenbank auslesen und Darstellen. Die Bestellung wird erst abgesendet, wenn der Shopbenutzer die AGB's akzeptiert hat (danach geht es bei darstellen = 4 weiter).
USER_BESTELLUNG_1.php	5
Beschreibung **DA**	Kreditkarten Handling *INTERN*: Kreditkartendaten Eingabemaske und Weiterleitung an darstellen = 7 für AGB Abfrage.
USER_BESTELLUNG_1.php	6
Beschreibung **DA**	Kreditkartenhandling *EXTERN*: Hier wird die Funktion payment_extern ($Pay) aufgerufen. Ihr wird im Argument eine Pay mitgegeben (siehe auch pay_def.php). Darin befinden sich alle nötigen Informationen, um eine Zahlung durchzuführen. Weiter geht es bei darstellen = 4.
USER_BESTELLUNG_1.php	7

 © 2001 R. Glanzmann & J. Fontanil

Modul: Aufruf-Module – Userseitiges Bestellungswesen Teil 2/2	
Beschreibung **DA**	Wenn wir eine interne Kreditkartenzahlung haben, so wird uns die Variable GNUPG mit dem Wert 1 von darstellen = 5 her übergeben. Wir überprüfen also zuerst die Kreditkarten Nummer. Falls sie falsch ist oder das Ablaufdatum ungültig ist, senden wir sie zurück und der Kunde soll sie nochmals eingeben. Zur Erkennung geben wir das Flag $Kreditkarten_Nummer_ungueltig = 1 mit. Wenn alles gut ging, geht es weiter bei darstellen = 4.
USER_BESTELLUNG_1.php	4
Beschreibung **DA**	- Unterscheidung des Workflows je nach Bezahlungsart: - Vorauskasse, Rechnung, Nachnahme: AGB akzeptiert?, weiter - Interne Kreditkartenzahlung: Nummer validieren, GNU-PG enable, weiter - Externe Kreditkartenzahlung: E-Mail String zurücklesen, weiter - Bestellung an Kunden und Shopinhaber mailen - Danke an Shopuser ausgeben (mit Kontaktierungsmöglichkeit bei Fragen) - Bestellung abschliessen - Alle alten Session-ID's aus Datenbank löschen

Modul: Aufruf-Module – pop_up.php / pop_up_admin.php	
Datei	**Zweck**
pop_up.php / pop_up_admin.php	
Beschreibung **DA**	Ein grosses Bild eines Artikels in einem Fenster darstellen. Mit der Möglichkeit, das Fenster per Link zu schliessen. Diese Funktion musste als eigene Datei ausprogrammiert werden (wegen HTML).

Modul: Aufruf-Module - Administrationsaufrufe	
Datei	**Variable $darstellen =**
SHOP_ADMINISTRATION_ AUFRUF.php	1
Beschreibung **DA**	Ein neuer, leerer Artikel wird erzeugt und an darstellenArtikel weitergegeben. Auf diese Weise kann man neue Artikel in die Datenbank aufnehmen.
SHOP_ADMINISTRATION_ AUFRUF.php	101
Beschreibung **DA**	Einem Artikel eine oder mehrere Kategorien zuordnen. Dies ist die Station nach der Eingabe der Artikeldaten. Man kommt von der Darstellen-Funktion darstellenArtikel($Artikel) hierhin. Von hier aus geht es weiter ins Modul bild_up.php, wo die Bilder zum Artikel gespeichert werden können.
SHOP_ADMINISTRATION_ AUFRUF.php	2

 © 2001 R. Glanzmann & J. Fontanil

Modul: Aufruf-Module - Administrationsaufrufe	
Beschreibung **DA**	Auswahlmenü (Kategorienwahl und Artikelwahl), wenn man einen Artikel zum Bearbeiten / Löschen auswählen will. Wenn man auf eine Kategorie / Unterkategorie klickt, baut sich das Fenster neu auf und man kann die sich darin befindenden Artikel sehen. Wenn man sich hier für einen entschieden hat, geht es weiter zu darstellen = 5.
SHOP_ADMINISTRATION_ AUFRUF.php	5
Beschreibung	Nachdem der Artikel bearbeitet wurde, wird hier entweder ein Update des Artikels oder das Löschen des Artikels eingeleitet. Beim Löschen wird die Funktion delArtikel ($Artikel_ID) aufgerufen, beim Bearbeiten eines Artikels wird updArtikel ($Artikel_ID) benutzt.
SHOP_ADMINISTRATION_ AUFRUF.php	6
Beschreibung **DA**	Eine neue Kategorie erstellen 1/2: Eingabe von Namen und der Positions-Nr der neuen (Haupt-)Kategorie. Beschreibung und Bilddaten blieben bei dieser Shop-Version noch aussen vor.
SHOP_ADMINISTRATION_ AUFRUF.php	7
Beschreibung **DA**	Eine neue Kategorie erstellen 2/2: Speichern der neuen (Haupt-) Kategorie.
SHOP_ADMINISTRATION_ AUFRUF.php	61
Beschreibung **DA**	Eine neue Unterkategorie erstellen Schritt 1/2: Eingabe der Unterkategoriedaten.
SHOP_ADMINISTRATION_ AUFRUF.php	71
Beschreibung **DA**	Eine neue Unterkategorie erstellen 2/2: Speichern der neuen Unterkategorie.
SHOP_ADMINISTRATION_ AUFRUF.php	20
Beschreibung **DA**	Eine Kategorie auf eine andere Position verschieben 1/2: Auswahl der neuen Position
SHOP_ADMINISTRATION_ AUFRUF.php	21
Beschreibung **DA**	Eine Kategorie auf eine andere Position verschieben 2/2: Speichern der neuen Position.
SHOP_ADMINISTRATION_ AUFRUF.php	25
Beschreibung **DA**	Eine Unterkategorie an eine andere Position innerhalb der gleichen Kategorie verschieben 1/2: Auswahl der neuen Position.
SHOP_ADMINISTRATION_ AUFRUF.php	26
Beschreibung **DA**	Eine Unterkategorie auf eine andere Position innerhalb der gleichen Kategorie verschieben 2/2: Speichern der neuen Position.

 © 2001 R. Glanzmann & J. Fontanil

Modul: Aufruf-Module - Administrationsaufrufe	
SHOP_ADMINISTRATION_ AUFRUF.php	30
Beschreibung **DA**	Eine Unterkategorie auf eine beliebige Stelle verschieben 1/2: Auswahl der neuen Position.
SHOP_ADMINISTRATION_ AUFRUF.php	31
Beschreibung **DA**	Eine Unterkategorie auf eine beliebige Stelle verschieben 2/2: Speichern der neuen Position.
SHOP_ADMINISTRATION_ AUFRUF.php	35
Beschreibung **DA**	Eine Kategorie umbenennen 1/2: Eingabe des neuen Kategorienamens.
SHOP_ADMINISTRATION_ AUFRUF.php	36
Beschreibung **DA**	Eine Kategorie umbenennen 2/2: Speichern der neuen Position.
SHOP_ADMINISTRATION_ AUFRUF.php	40
Beschreibung **DA**	Eine Unterkategorie umbenennen 1/2: Eingabe des neuen Namens.
SHOP_ADMINISTRATION_ AUFRUF.php	41
Beschreibung **DA**	Eine Unterkategorie umbenennen 2/2: Speichern der neuen Position
SHOP_ADMINISTRATION_ AUFRUF.php	45
Beschreibung **DA**	Eine Kategorie/Unterkategorie löschen 1/2: Bestätigungsanfrage!
SHOP_ADMINISTRATION_ AUFRUF.php	46
Beschreibung **DA**	Eine Kategorie/Unterkategorie löschen 2/2: Löschvorgang ausführen!

Modul: Aufruf-Module – Bild in die Datenbank schreiben	
Datei	**Variablenwerte**
bild_up.php	Variable $Speichern = true

 © 2001 R. Glanzmann & J. Fontanil

Modul: Aufruf-Module – Bild in die Datenbank schreiben	
Beschreibung DA	Das bedeutet, der Shop-Administrator will ein Bild zu diesem Artikel in der Datenbank speichern. Nun wird ein neues Thumbnail vom eingelesenen Bild angefertigt (aus lizenzrechtlichen Gründen kein GIF-Support) und alle Bilddaten (insgesamt 4 Attribute der Tabelle artikel) in der Datenbank gespeichert. Diese Funktionalität verlangt seit v.1.05 *nicht* mehr nach root-Rechten.
bild_up.php	Variable $loeschen_Bild = 1, $Speichern = false
Beschreibung	Der Shop-Administrator hat sich in diesem Fall entschieden, das vorhandene Bild zu löschen. Dies wird nun ausgeführt.

Modul: Aufruf-Module – Bestellungsmanagement	
Datei	**$darstellen =**
SHOP_BESTELLUNG.php	10
Beschreibung DA	Kundendaten und seine Bestellung der Referenznummer nach anzeigen. (Hier kann man Änderungen am Kundenprofil vornehmen / die Bestellung löschen).
SHOP_BESTELLUNG.php	11
Beschreibung DA	Einen zuvor gewählten Artikel aus der aktuell bearbeiteten Bestellung löschen.
SHOP_BESTELLUNG.php	12
Beschreibung DA	Die aktuelle Bestellung löschen oder sie updaten (abspeichern). Wenn die Variable $Speichern gleich "Speichern" ist, soll upgedated werden, sonst wird gelöscht.
SHOP_BESTELLUNG.php	13
Beschreibung DA	Hier werden alle abgeschlossenen aber noch nicht gelöschten Bestellungen tabellarisch angezeigt. Per Klick auf eine Zeile kann die Bestellung editiert werden.
SHOP_BESTELLUNG.php	*ungleich 10, 11, 12, 13*
Beschreibung DA	Suchformular anzeigen. Momentan kann man hier nur nach Referenznummern suchen.

Modul: Aufruf-Module – Kundenattribute Management	
Datei	**$darstellen =**
SHOP_KUNDE.php	10
Beschreibung DA	Abspeichern der Einstellungen.
SHOP_KUNDE.php	*ungleich 10*
Beschreibung DA	Formular für das Mutieren der Kundenattribute (Datenfelder eines Kunden-Datensatzes) anzeigen. Hier können auch die selbstdefinierten Zusatzfelder konfiguriert werden.

Modul: Aufruf-Module – Layout Management	
Datei	**$darstellen =**

 © 2001 R. Glanzmann & J. Fontanil

Modul: Aufruf-Module – Layout Management	
SHOP_LAYOUT.php	10
Beschreibung **DA**	Speichert die veränderten Layout-Parameter in die Datenbank und ruft die Funktionen zur Erzeugung der CSS-Dateien, sowie von index.php (beinhaltet Frameset) auf.
SHOP_LAYOUT.php	20
Beschreibung **DA**	Gibt ein Formular aus, in welchem man per Browse-Button eine Bild-Datei auf der Festplatte auswählen kann. Diese wird dann als eines der folgenden Bilder hochgeladen: Shoplogo, linkes-, oberes-, content-Hintergrundbild.
SHOP_LAYOUT.php	21
Beschreibung **DA**	Dies ist die Folgeseite von darstellen = 20. Hier wird der Dateiname der hochgeladenen Datei analysiert. Handelt es sich um ein File vom Typ .gif, .jpg, jpeg oder .png so wird sie in die entsprechende Bilddatei im Verzeichnis "Bilder" hineinkopiert. Auch für diese Bild-Funktion werden *keine* root-Rechte mehr benötigt.
SHOP_LAYOUT.php	30
Beschreibung **DA**	Gibt ein Formular aus, in welchem man per Browse-Button eine Bild-Datei auf der Festplatte auswählen kann, mit welcher man einen Shopbutton ersetzen kann.
SHOP_LAYOUT.php	31
Beschreibung **DA**	Dies ist die Folgeseite von darstellen = 30. Analysiert wird der Dateiname des hochgeladenen Files. Handelt es sich um eine Datei vom Typ .gif, so wird der gewählte Button ersetzt (gespeichert). Die Analyse erfolgt in JavaScript.
SHOP_LAYOUT.php	*ungleich 10, 20, 21, 30, 31*
Beschreibung **DA**	Gibt die Eingabemaske für die Layout Settings als Formular aus (mit den momentan eingestellten Werten aus der Datenbank eingefüllt).

Modul: Aufruf-Module – Allgemeine Shopeinstellungen	
Datei	**$darstellen =**
SHOP_SETTINGS.php	10
Beschreibung **DA**	Weiche um zu unterscheiden, ob das Formular abgeschickt wurde (Speichern-Button). Shop-Settings aufbereiten und abspeichern.
SHOP_SETTINGS.php	*ungleich 10*
Beschreibung **DA**	Formular zum Editieren der Shop-Settings.

Modul: Aufruf-Module – Versandkosten Management	
Datei	**$darstellen =**
SHOP_VERSANDKOSTEN.php	10
Beschreibung **DA**	Hier werden die Versandkostensettings gespeichert (Funktion: setversandkostensettings(..)) und es wird eine Erfolgsmeldung ausgegeben.

© 2001 R. Glanzmann & J. Fontanil

Modul: Aufruf-Module – Versandkosten Management	
SHOP_VERSANDKOSTEN.php	*ungleich 10*
Beschreibung **DA**	Formular in welchem man die Versandkostensettings editieren kann.

Modul: Darstellungs-Module - Bestellung	
Datei	**Funktionsname**
USER_BESTELLUNG_ DARSTELLUNG.php	darstellenBestellung
Beschreibung **DA**	Diese Funktion stellt die im Argument gelieferte Bestellung in HTML dar und leitet die Daten dann zur Übertragung in die Datenbank weiter. Ist der übergebene löschen-Wert true, dann wird die Möglichkeit angezeigt, Artikel aus dem Warenkorb zu entfernen (ist bei der Darstellung des Warenkorbes zusammen mit den Adressdaten der Bestellung nicht mehr erwünscht!). Ist das Flag Admin = true, so werden für die Buttons andere Pfade benutzt. Damit die Bestellung für immer statisch bleibt und die Preise nicht jedes Mal neu berechnet werden, wenn man im Bestellungsmanagement die Bestellungen editiert, musste noch ein weiteres Flag eingebaut werden, welches die Funktion berechneversandkosten(..) ausklammert. **Argumente**: Bestellungs-Objekt, löschen-Flag, Administrator-Flag **Rückgabewert**: Darstellung des Warenkorbes in HTML
USER_BESTELLUNG_ DARSTELLUNG.php	darstellenStringBestellung
Beschreibung **DA**	Diese Funktion schreibt den Warenkorb-Inhalt formatiert in einen String, der zum Versand der E-Mails an den Shopbenutzer und -betreiber gebraucht wird. Funktionalität ansonsten ähnlich wie Funktion 'darstellenBestellung'. **Argumente**: Bestellungs-Objekt **Rückgabewert:** Abgefüllte Bestellung in einem *String*

Modul: Darstellungs-Module - Artikel	
Datei	**Funktionsname**
SHOP_ADMINISTRATION _ARTIKEL.php	darstellenArtikel
Beschreibung **DA**	Diese Funktion stellt den im Argument gelieferten Artikel dar und leitet die Daten dann zur Übertragung in die Datenbank weiter an die Kategorien- und die Bild-Eintragung. Am Ende des Eingabeformulars werden noch Buttons (Reset, Submit, Abbrechen, Hilfe) zur Navigation dargestellt. **Argumente**: Ein Artikel als Artikel-Objekt (Definition siehe artikel_def.php) **Rückgabewert**: HTML-Teil einer Page (Formular, ohne Head und ohne Body-Tags)

 © 2001 R. Glanzmann & J. Fontanil

Modul: Darstellen Funktionen – Bilder aus der Datenbank auslesen	
Datei	**Funktionsname**
bild_view.php	bild_view
Beschreibung DA	Diese Datei wird nur innerhalb des HTML-Tags <IMG src="bild_view.php?..."...> verwendet. Es liest aus der Datenbank das entsprechende Bild aus und stellt es dann bereit. Die Datei ruft ihre einzige Funktion selbst auf. **Argumente**: Artikel_ID (INT), Bildformat (String), Bildgrösse (enum('gross', 'klein')) **Rückgabewert**: Bild

Modul: Darstellen Funktionen – Hilfe aus der Datenbank auslesen	
Datei	**Funktionsname**
USER_ADMIN_HILFE.php, ADMIN_HILFE.php	get_Hilfe
Beschreibung	Diese Funktion liefert einen String (inkl. HTML-Formatierung), der den entsprechenden Hilfetext der angeforderten Seite beinhaltet. Die zu übergebende Hilfe_ID besteht aus dem Namen der anfordernden Datei und hat angehängt einen Suffix von _1, _2... **Argumente**: Hilfe_ID **Rückgabewert**: Hilfetext zur aktuellen Seite (HTML-String)

Modul: Backup Backup Menü / Backup / Restore	
Datei	**Funktionalität**
SHOP_BACKUP.php, SHOP_BACKUP_f1.php, ADMIN_backup.php, ADMIN_restore.php [, index.php]	Backup und Restore Untermenü [automatisiertes] Datenbank Backup Restore / Download eines Datenbank Backups
Beschreibung DA	Die Datenbank Backuplösung des PhPepperShops verteilt sich auf vier Dateien. SHOP_BACKUP und SHOP_BACKUP_f1.php stellen dem Benutzer ein Backup Untermenü und eine Backup Konfigurationsmaske zur Verfügung. Die eigentliche Backup-Funkionalität und die Restore-Maske befinden sich in den Dateien ADMIN_backup.php und ADMIN_restore.php. Wurde das Backup als periodisch konfiguriert, wird die Datei index.php angepasst (Der String liegt in der Datenbank in der Tabelle css_file unter der ID `backup`).

 © 2001 R. Glanzmann & J. Fontanil

Modul: Darstellen Funktionen – Payment Schnittstelle	
Datei	**Funktionsname**
payment_interface.php	payment_extern
Beschreibung DA	Diese Funktion liefert nicht im üblichen Sinne (mittels Return-Wert) einen Rückgabewert zurück, sondern gibt direkt mittels des `echo` Kommandos auf das Webdokument ein Formular aus. Dieses Formular beinhaltet, wie oben erwähnt, hidden-Fields, welche die Daten zur Zahlung beinhalten. Will man weitere Daten an das externe Zahlungssystem senden, kann man diese aus dem übergebenen Pay-Objekt herausholen und in einem weiteren hidden-Field dem externen Zahlungssystem mit übergeben. Um zu verstehen, was alles in einem Pay-Objekt vorhanden ist, siehe folgende Klassendefinitionen oder die Beschreibung im Kapitel 2.8.2): --> pay_def.php --> kunde_def.php --> bestellung_def.php --> artikel_def.php --> kategorie_def.php Diese Funktion besteht im wesentlichen aus vier Teilen: 1.) Definition des URLs des externen Zahlungssystems. Auf diese Adresse wird gesprungen, wenn der Kunde die allgemeinen Geschäftsbedingungen akzeptiert. 2.) Auspacken der benötigten Variablen aus dem übergebenen Pay-Objekt 3.) Abfüllen der zu übergebenen Werte in ihre hidden-Fields 4.) Rückgabewert auf true setzen **Argumente**: Pay-Objekt **Rückgabewert**: true bei Erfolg, sonst Abbruch mit der PHP-Funktion 'die'

Modul: Administrations Menü	
Datei	**Aufruf**
Shop_Einstellungen_Menu_1.php	
 DA	SHOP_ADMINISTRATION_AUFRUF.php?darstellen=1
	SHOP_ADMINISTRATION_AUFRUF.php?darstellen=2&up_loe=1
	SHOP_ADMINISTRATION_AUFRUF.php?darstellen=2&up_loe=0
	Shop_Einstellungen_Menu_Kategorien.php
	SHOP_BESTELLUNG.php
	SHOP_SETTINGS.php
	SHOP_LAYOUT.php
	SHOP_LAYOUT.php?darstellen=20
	SHOP_LAYOUT.php?darstellen=30
	SHOP_VERSANDKOSTEN.php
	SHOP_BACKUP.php
	index.html
	USER_ADMIN_HILFE.php?Hilfe_ID=Shop_Einstellungen_Menu_1

 © 2001 R. Glanzmann & J. Fontanil

Modul: HTML-Darstellung	
Datei	
index.html / index.php	
Beschreibung **DA**	Das `index.html` leitet direkt auf `index.php` weiter. Dort wird dem Surfer bereits eine Session und somit eine leere Bestellung zugeordnet. Nun kann er den Shop vollumfänglich benutzen. Die Datei `index.php` baut noch das Frameset.
top.php / content.html	
Beschreibung **DA**	Bilden des Framesets (FRAME = 'left' wird von einer Aufruf-Funktion mit Inhalt gefüllt). Das File `content.html` hält einen Willkommens-Bildschirm bereit. top.php zeigt den Shop-Namen an und bietet dem Kunden Navigationselemente wie Warenkorb, Kasse und Hilfe. Ausserdem versteckt sich dort hinter dem kleinen Stern der Passwort geschützte Eingang zur Shop-Administration.

2.2.3 Fehlerbehandlung in den Modulen

Allgemeine Sicherheitsvorkehrungen

Funktionen geben immer einen Rückgabewert, welcher entweder definiert ist oder aber true ist.

Wenn eine Funktion mit der Datenbank kommuniziert, wird nach jeder Verbindung überprüft, ob die Kommunikation erfolgreich durchgeführt werden konnte.

Fehlercode

Entsteht in einer Funktion ein Fehler, so bricht diese mit dem PHP-Befehl 'die("Meldung")' ab. Die Meldung ist speziell formatiert: Zuerst wird der **Modul-Code** ausgegeben. Er setzt sich aus den ersten Buchstaben der Modulnamen zusammen. Die **Meldung** enthält eine genauere Beschreibung des Fehlers, des Ortes oder des Zustandes von lokalen Variablen. Danach folgt, meist in runden Klammern geschrieben, der **Name der Funktion**, in welcher der Fehler aufgetreten ist. Optional ist der vierte Teil einer Fehlermeldung. Er wird meist nur bei komplexeren oder grösseren Funktionen verwendet. Hier werden **funktionsinterne Angaben** mitgegeben, z.B. wo in der Funktion der Fehler aufgetreten ist, welches SQL den Fehler verursacht hat oder ähnliches.

Bei Fehlern mit SQL-Queries geben die Funktionen zudem auch die Query selbst noch aus. Auch der Datenbank-Wrapper (database.php) wird in einem Fehlerfall die nicht erfolgreiche Query zur Sicherheit nochmals anzeigen.

Beispiele

Modul-Code	*Meldung*	*Funktion*	*Funktionsintern*
U_B_Error	Konnte die Bestellung nicht dem Kunden zuweisen	gibBestellung_an_Kunde	-
S_A_Error	(RS_ukat) Eine Unterkategorie konnte nicht entfernt werden	delKategorie	-

 © 2001 R. Glanzmann & J. Fontanil

Modul-Code	*Meldung*	*Funktion*	*Funktionsintern*
S_A_Error	RS ist kein Objekt->Abbruch	katposschieben	Del_Kat_1_1

U_B_Error kommt also von USER_BESTELLUNG.php, S_A_Error kommt von SHOP_ADMINISTRATION.php. Bei USER_BESTELLUNG_DARSTELLUNG.php würde man U_B_D_Error schreiben. Wir haben aber auch an Namenskonflikte gedacht. bei SHOP_ADMINISTRATION_AUFRUF.php und SHOP_ADMINISTRATION_ARTIKEL.php wird der uneindeutige Buchstabe einfach ausgeschrieben.

 © 2001 R. Glanzmann & J. Fontanil

2.3 Session Management

Das Session-Management wird über die PHP4-interne Session-Verwaltung abgewickelt und wurde grundsätzlich ohne Veränderung aus der Projektarbeit übernommen.

PHP4 ist grundsätzlich in der Lage, die Verknüpfung der Zugriffe über URL oder Cookies vorzunehmen. Die Session Konfiguration läuft weitgehend über die Datei `php.ini` ab (bei uns im Verzeichnis `/etc/`). Dort beliessen wir alle Session relevanten Einstellungen auf den Default-Werten. Auf diese Weise konfiguriert, arbeitet unser Session-Management mit Cookies, deren Lebenszeit auf die Laufzeit des Browsers beschränkt ist.

2.3.1 Maximale Session Dauer

Weiter wird in der `php.ini` die maximale Zeit einer Session auf 1440 Sekunden, also rund 24 Minuten, beschränkt (Default Einstellung). In den Shop-Settings kann der Shop-Administrator aber die Zeit einer Session verändern. Was immer er in diesem Feld eingibt, wird nur zum Tragen kommen, solange die dort angegebene maximale Session-Zeit kürzer oder gleich lang ist, als die in der `php.ini`. Weiter ist zu beachten, dass die Zeit, die in den Shop-Settings eingestellt ist, sich nicht “direkt“ auf die Session-Variable auswirkt.

Es gibt PHP-Funktionen mit welchen man die Variablen in der php.ini ändern könnte und somit auch die obere Limite flexibler setzen könnte. Auf den Einsatz einer solchen Funktion haben wir aber verzichtet, da kaum ein Provider Freude darüber empfindet, wenn seine Kunden in seiner `php.ini` Änderungen vornehmen könnten. Ausserdem hat man als Provider-Kunde keinen schreibenden Zugriff auf die `php.ini`.

Wenn ein Kunde den Shop betritt, erhält er nicht nur eine Session-ID, sondern auch gleich eine leere Bestellung. Dort drin gibt es ein Feld namens `expired`. Darin wird die Summe aus der aktuellen Zeit und der in den Shop-Settings definierten maximalen Session-Dauer abgespeichert. Bei jeder neuen Seite, die den Session-Management-Header (siehe weiter unten) beinhaltet, wird geprüft, ob diese Zeit nicht schon abgelaufen ist. Ist dies der Fall, so wird die Bestellung in der Datenbank gelöscht und der Kunde erhält eine neue, leere Bestellung.

Rekapitulierend kann man also sagen: Der Kunde erhält zwar keine neue Session-ID, er erhält aber eine neue Bestellung, was für ihn zum gleichen Resultat führt. Er wird vom Shopsystem als 'neuer' Kunde gesehen.

2.3.2 Session-Management-Header

Am Ende dieses Kapitels findet man den Code, den wir Session-Management-Header nennen. Er ist überall dort in die Files integriert worden, wo der Kunde vom Shop einen HTML-Header (also eine neue Seite) zugeschickt bekommt.

Dieser Header schaut, ob die Session-ID, bei uns $mySession_ID genannt, schon vorhanden ist. Dann soll die Session-ID Ausgabefunktion diese Variable auslesen. Falls die Variable noch nicht existiert, soll unsere Session_ID $mySession_ID heissen, diese Variable an die aktuelle Session gebunden werden und eine Session eröffnet werden.

Als einzige Änderung am Session Management während Diplomarbeit, mussten wir die Funktionen schliessenBestellung(..), delallexpiredSessions(..), delSession(..) erweitern. Da wir jetzt auch noch eine Kunden- und eine Kunden-Referenz Tabelle haben, muss beim Löschen einer Bestellung auch ein temporärer Kunde gelöscht werden (siehe auch Kapitel 2.11, CRM).

 © 2001 R. Glanzmann & J. Fontanil

```
// ---------------------------------------------------------------------
// Session_Management: (Ueberpruefen ob eine Session_ID uebergeben wurde,
// sonst eine Neue erzeugen
if(empty($mySession_ID)){
    session_name("mySession_ID");
    if (!session_register("mySession_ID")) {
        die("<HTML><BODY><P><H1>USER_BESTELLUNG_AUFRUF.php: ACHTUNG session_regis
             ter() = false, konnte keine Session_ID zuteilen</H1></P><BR><BR></BODY>
             </HTML>");
    }
}
else {
    session_id($mySession_ID);
}
```

 © 2001 R. Glanzmann & J. Fontanil

2.4 Datenbank

Das Datenbankdesign ist eine sehr zentrale Komponente bei der Entwicklung eines Web-Shop-Frameworks. Dementsprechend investierten wir viel Zeit und Energie in die fortlaufende Entwicklung dieses Designs. *Security-Aspekte* zum Datenbankdesign wurden schon im Kapitel 2.1 behandelt. Wir konnten auf einem guten Fundament aus der Praktischen Arbeit aufbauen.

2.4.1 Idee und Konzept

Damit wir einen möglichst dynamischen Shop, also einen Shop mit vielen Freiheiten und grossem Einsatzgebiet erhalten, haben wir uns schon in der Praktischen Arbeit für ein etwas komplexeres Design entschieden.

Weiter war es eine Grundidee, dass man den Shop *komplett* in der Datenbank hat. Dies ist vom Backup her ein Vorteil (einfach Backup der Datenbank anfertigen, fertig) und vom Zugriff her, dachten wir, auch. Leider haben wir uns vor dem Design wenig mit BLOBs in Datenbanken beschäftigt. Es erwies sich später als hinderlich, wenn man BinaryLargeObjects in der gleichen Tabelle hat wie weitere Daten. Es ist uns bewusst, dass man saubererweise die Bilder der Artikel in einer gesonderten Tabelle hätte ablegen müssen. Wir trugen dieser Erkenntnis Rechnung und haben alle weiteren Bilder (Layout Management) im Dateisystem abgelegt.

Das ursprüngliche Datenbankdesign wurde um einige Tabellen und viele Attribute ergänzt. Zum Beispiel haben wir aufgrund des CRMs die Kundendaten aus der Bestellung ausgegliedert und ihnen eine eigene Tabelle (kunde) gegeben. Bei dieser Gelegenheit haben wir auch den Kundendatensatz überarbeitet und ihm weitere Attribute gegeben (Postfach,...). Wir haben versucht, alle möglichen Adressdaten zu berücksichtigen. Dank der neuen, dynamischen Kundenattribut-Verwaltung kann man jetzt auch flexibel auswählen, mit welchen Kundenattributen man arbeiten will.

2.4.2 Personalisierung durch das Installationstool config.pl

Damit mehrere Shops gleichzeitig in einem DBMS koexistieren können, müssen die Datenbanken der einzelnen Shops personalisiert sein. Der alte Shop der Praktischen Arbeit arbeitete immer mit der Datenbank namens shopdb. Mühsam musste bei jeder Installation der Shopname und der Datenbankuser angepasst werden. Dank dem Installationsscript `config.pl` wird einem diese Arbeit jetzt abgenommen.

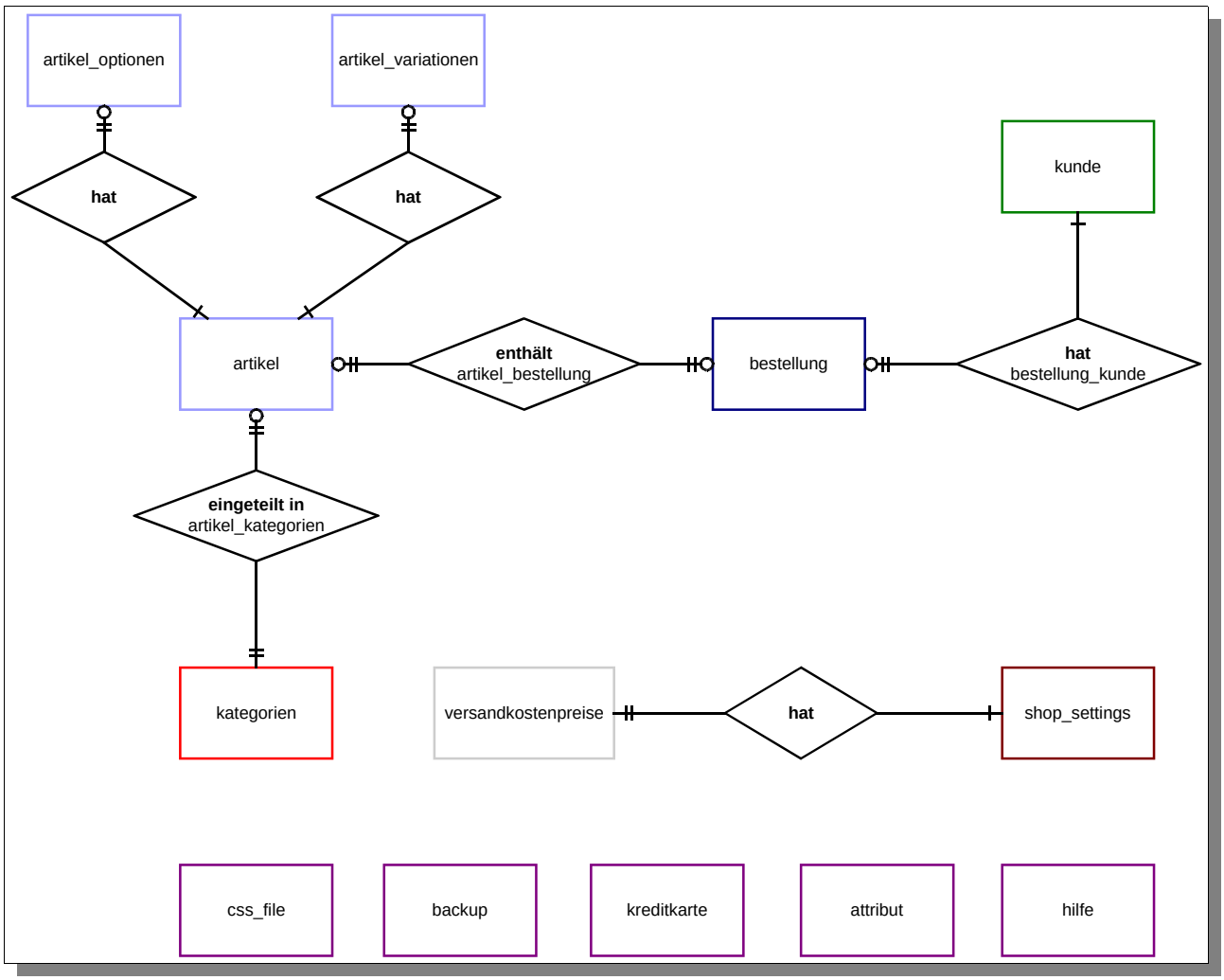
Das Perl-Script `config.pl` erstellt folgende personalisierten Datenbank Dateien (SQL):

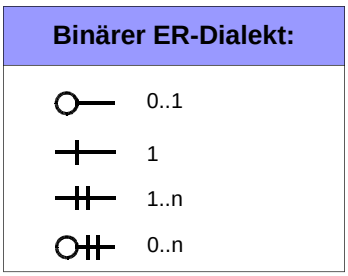
- `<shopname>_create.sql` Datenbank und User erstellen
- `<shopname>_inser.sql` Default-Shopdaten in die Datenbank laden
- `<shopname>_del.sql` Datenbank und User löschen

Personalisiert wird hierbei gespeichert, ob User angelegt werden sollen oder nicht und wie der Name der Datenbank lautet.

 © 2001 R. Glanzmann & J. Fontanil

2.4.3 Relationen in der Datenbank (Übersicht)





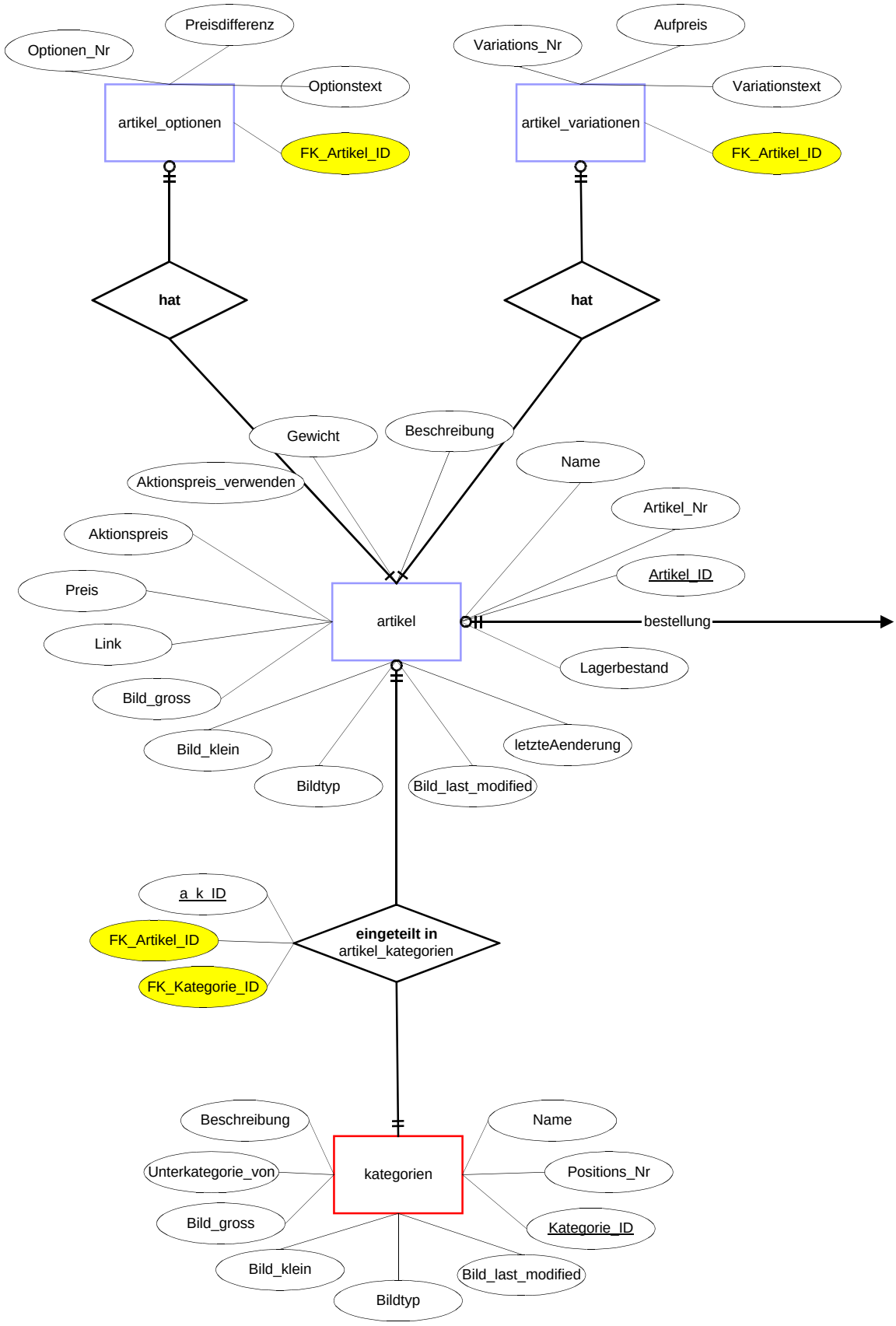
2.4.4 Entity Relationship Diagramm der Shop Datenbank

Das Entity Relationship Diagramm der PhPepperShop Datenbank wird hier inklusive allen Tabellen-Attributen dargestellt. Die Darstellung wurde so gewählt, dass die Datenbank in drei Teile aufgesplittet gezeigt wird.

Im Teil 1 sieht man den Artikel mit seinen Optionen und Variationen und der 'm zu n'-Verknüpfung zu den Kategorien. Teil 2 zeigt die Beziehung zwischen einem Kunden und seinen Bestellungen. Der dritte Teil ist losgelöst von den anderen beiden und stellt die Shop-Settings mit den Versandkostenpreisen und allen weiteren Zusatztabellen dar.

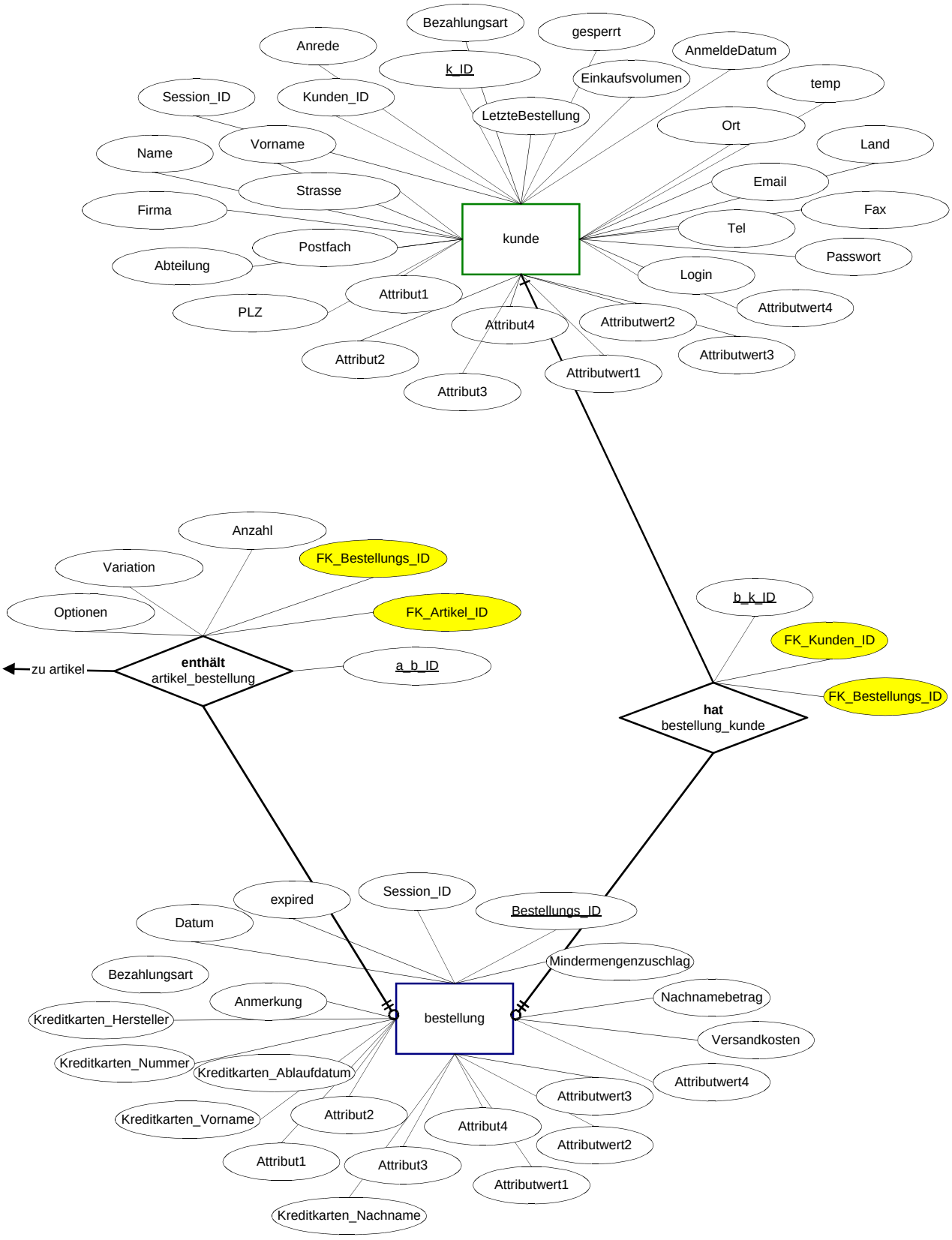
 © 2001 R. Glanzmann & J. Fontanil

ER Diagramm Teil 1



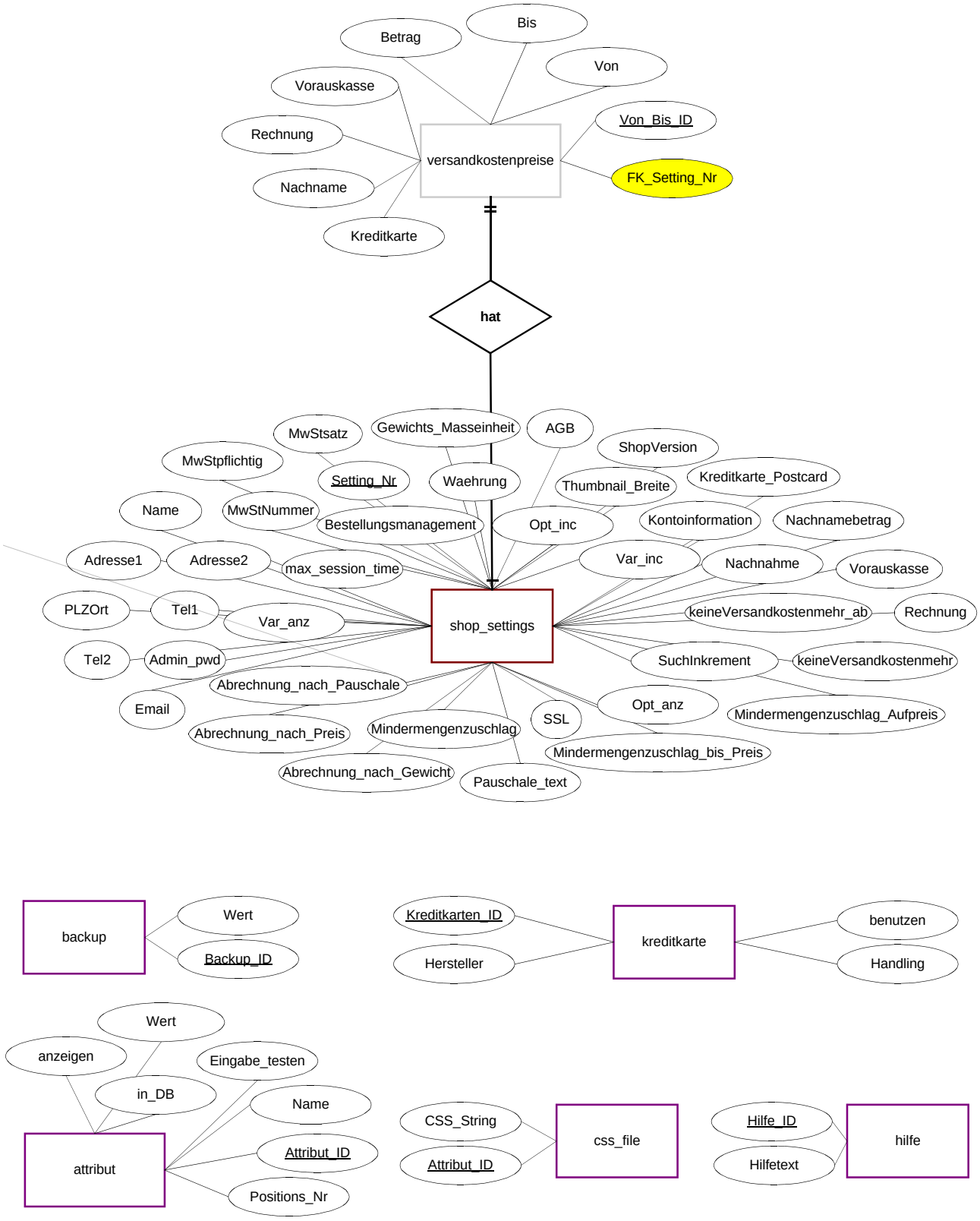
 © 2001 R. Glanzmann & J. Fontanil

ER Diagramm Teil 2



 © 2001 R. Glanzmann & J. Fontanil

ER Diagramm Teil 3



 © 2001 R. Glanzmann & J. Fontanil

2.4.5 Tabellen und ihre Attribute im Detail

Es folgen nun die Tabellen im Detail. Hier sieht man auch welche Datentypen für die jeweiligen Attribute verwendet wurden:

artikel

```
Artikel_ID INT NOT NULL DEFAULT 0 AUTO_IN-
CREMENT,
Artikel_Nr VARCHAR(128) NOT NULL,
Name VARCHAR(128) NOT NULL,
Beschreibung TEXT NOT NULL,
Preis DOUBLE NOT NULL DEFAULT 0.0,
Aktionspreis DOUBLE NOT NULL DEFAULT 0.0,
Aktionspreis_verwenden enum('N','Y') NOT NULL
                        DEFAULT 'N',
Gewicht DOUBLE NOT NULL DEFAULT 0.0,
Link VARCHAR(255),
Bild_gross MEDIUMBLOB,  #/* Achtung: Bei anderer
                        DB, Typ ev. = BLOB */
Bild_klein BLOB,
Bildtyp VARCHAR(15),
Bild_last_modified TIMESTAMP(14),
Lagerbestand INT NOT NULL DEFAULT 0,
letzteAenderung DATE,
PRIMARY KEY (Artikel_ID),
UNIQUE UC_Artikel_ID (Artikel_ID)
```

kategorien

```
Kategorie_ID INT NOT NULL DEFAULT 0
                            AUTO_INCREMENT,
Positions_Nr INT NOT NULL DEFAULT 0,
Name VARCHAR(128) NOT NULL,
Beschreibung TEXT NOT NULL,
Bild_gross MEDIUMBLOB,  #/* Achtung: Bei anderer
                        DB, Typ ev. = BLOB */
Bild_klein BLOB,
Bildtyp VARCHAR(15),
Bild_last_modified TIMESTAMP(14),
Unterkategorie_von VARCHAR(128) DEFAULT NULL,
PRIMARY KEY (Kategorie_ID),
UNIQUE UC_Kategorie_ID (Kategorie_ID),
INDEX IDX_kategorien_1 (Name)
```

bestellung

```
Bestellungs_ID INT NOT NULL DEFAULT 0 AUTO_IN-
CREMENT,
Session_ID CHAR(32) NOT NULL DEFAULT 0,
# /* Unsere getesteten Session_IDs waren immer
CHAR(32)... aber man weiss ja nie */
expired BIGINT NOT NULL DEFAULT 0,
# /* In andernen DBs ev. anderer Typ (10 Stel-
len) */
Datum Date,
Anmerkung TEXT,
Bezahlungsart CHAR(32) NOT NULL,
Kreditkarten_Hersteller CHAR(32) NOT NULL,
Kreditkarten_Nummer CHAR(16) NOT NULL,
Kreditkarten_Ablaufdatum VARCHAR(10) NOT NULL,
Kreditkarten_Vorname VARCHAR(128) NOT NULL,
Kreditkarten_Nachname VARCHAR(128) NOT NULL,
Attribut1 VARCHAR(128) NOT NULL,
Attribut2 VARCHAR(128) NOT NULL,
Attribut3 VARCHAR(128) NOT NULL,
Attribut4 VARCHAR(128) NOT NULL,
Attributwert1 VARCHAR(128) NOT NULL,
Attributwert2 VARCHAR(128) NOT NULL,
Attributwert3 VARCHAR(128) NOT NULL,
Attributwert4 VARCHAR(128) NOT NULL,
Versandkosten DOUBLE NOT NULL DEFAULT 0.0,
Nachnamebetrag DOUBLE NOT NULL DEFAULT 0.0,
Mindermengenzuschlag DOUBLE NOT NULL,
temp_message_string TEXT NOT NULL,
Rechnungsbetrag DOUBLE NOT NULL DEFAULT 0.0,
                        DEFAULT 0.0,
Bestellung_abgeschlossen enum('N','Y')
                        NOT NULL DEFAULT 'N',
Bestellung_ausgeloest enum('N','Y')
                        NOT NULL DEFAULT 'N',
Bestellung_bezahlt enum('N','Y')
                        NOT NULL DEFAULT 'N',
PRIMARY KEY (Bestellungs_ID)
UNIQUE UC_Bestellungs_ID (Bestellungs_ID)
```

kunde

```
k_ID INT NOT NULL DEFAULT 0 AUTO_INCREMENT,
Kunden_ID VARCHAR(255) NOT NULL,
Kunden_Nr VARCHAR(128) NOT NULL DEFAULT 0,
Session_ID CHAR(32) NOT NULL DEFAULT 0,
Anrede VARCHAR(24) NOT NULL,
Vorname VARCHAR(128) NOT NULL,
Nachname VARCHAR(128) NOT NULL,
Firma VARCHAR(128) NOT NULL,
Abteilung VARCHAR(128) NOT NULL,
Strasse VARCHAR(128) NOT NULL,
Postfach VARCHAR(16) NOT NULL,
PLZ VARCHAR(32) NOT NULL,
Ort VARCHAR(128) NOT NULL,
Land VARCHAR(128) NOT NULL,
Tel VARCHAR(32) NOT NULL,
Fax VARCHAR(32) NOT NULL,
Email VARCHAR(128) NOT NULL,
Login VARCHAR(32) NOT NULL,
Passwort VARCHAR(32) NOT NULL,
gesperrt enum('N','Y') NOT NULL DEFAULT 'N',
temp enum('N','Y') NOT NULL DEFAULT 'Y',
Bezahlungsart VARCHAR(32) NOT NULL,
AnmeldeDatum Date,
LetzteBestellung Date,
Einkaufsvolumen DOUBLE NOT NULL DEFAULT 0.0,
#/* Attribute sind selbstkonfigurierbare Ein-
traege */
Attribut1 VARCHAR(128) NOT NULL,
Attribut2 VARCHAR(128) NOT NULL,
Attribut3 VARCHAR(128) NOT NULL,
Attribut4 VARCHAR(128) NOT NULL,
Attributwert1 VARCHAR(128) NOT NULL,
Attributwert2 VARCHAR(128) NOT NULL,
Attributwert3 VARCHAR(128) NOT NULL,
Attributwert4 VARCHAR(128) NOT NULL,
PRIMARY KEY (k_ID),
UNIQUE UC_k_ID (k_ID),
UNIQUE UC_Kunden_ID (Kunden_ID),
UNIQUE UC_Login (Login),
INDEX IDX_Kunden_ID_1 (Kunden_ID)
```

 © 2001 R. Glanzmann & J. Fontanil

attribut

```
Attribut_ID INT NOT NULL DEFAULT 0
                        AUTO_INCREMENT,
Name VARCHAR(128) NOT NULL,
Wert VARCHAR(128) NOT NULL,
anzeigen enum('N','Y') NOT NULL DEFAULT 'Y',
in_DB enum('N','Y') NOT NULL DEFAULT 'Y',
Eingabe_testen enum('N','Y')
                        NOT NULL DEFAULT 'Y',
Positions_Nr INT NOT NULL DEFAULT 0,
PRIMARY KEY (Attribut_ID),
UNIQUE UC_Attribut_ID (Attribut_ID),
UNIQUE UC_Positions_Nr (Positions_Nr)
```

artikel_kategorie

```
a_k_ID INT NOT NULL DEFAULT 0 AUTO_INCREMENT,
FK_Artikel_ID INT NOT NULL,
FK_Kategorie_ID INT NOT NULL,
FOREIGN KEY (FK_Artikel_ID)
          REFERENCES artikel (Artikel_ID),
FOREIGN KEY (FK_Kategorie_ID)
          REFERENCES kategorien (Kategorie_ID),
PRIMARY KEY (a_k_ID)
```

artikel_optionen

```
Optionen_Nr INT NOT NULL DEFAULT 1,
Optionstext CHAR(64),
Preisdifferenz DOUBLE NOT NULL DEFAULT 0.0,
FK_Artikel_ID INT NOT NULL,
FOREIGN KEY (FK_Artikel_ID)
            REFERENCES artikel (Artikel_ID)
```

artikel_variationen

```
Variations_Nr INT NOT NULL DEFAULT 1,
Variationstext CHAR(64),
Aufpreis DOUBLE NOT NULL DEFAULT 0.0,
FK_Artikel_ID INT NOT NULL,
FOREIGN KEY (FK_Artikel_ID)
            REFERENCES artikel (Artikel_ID)
```

artikel_bestellung

```
a_b_ID INT NOT NULL DEFAULT 0 AUTO_INCREMENT,
FK_Artikel_ID INT NOT NULL,
FK_Bestellungs_ID INT NOT NULL,
Anzahl INT NOT NULL DEFAULT 1,
Variation TEXT NOT NULL,
#/* Aufbau einer Variation:  Variationstext þ
Aufpreis */
Optionen TEXT NOT NULL,
#/* Aufbau einer Variation:  Optionstext þ
Preisdifferenz ... */
FOREIGN KEY (FK_Artikel_ID)
       REFERENCES artikel (Artikel_ID),
FOREIGN KEY (FK_Bestellungs_ID)
       REFERENCES bestellung (Bestellungs_ID),
PRIMARY KEY (a_b_ID,FK_Artikel_ID,FK_Bestel
       lungs_ID)
#/* Anm. Das Trennzeichen ist ALT+0254 */
```

bestellung_kunde

```
b_k_ID INT NOT NULL DEFAULT 0 AUTO_INCREMENT,
FK_Kunden_ID VARCHAR(255) NOT NULL,
FK_Bestellungs_ID INT NOT NULL,
FOREIGN KEY (FK_Kunden_ID)
          REFERENCES kunde (Kunden_ID),
PRIMARY KEY (b_k_ID)
```

hilfe

```
Hilfe_ID VARCHAR(128) NOT NULL,
Hilfetext TEXT,
PRIMARY KEY (Hilfe_ID),
UNIQUE UC_Hilfe_ID (Hilfe_ID)
```

css_file

```
Attribut_ID CHAR(32) NOT NULL,
CSS_String CHAR(255),
PRIMARY KEY (Attribut_ID),
UNIQUE UC_Attribut_ID (Attribut_ID)
```

versandkostenpreise

```
Von_Bis_ID INT DEFAULT 0 NOT NULL
          AUTO_INCREMENT,
Von DOUBLE DEFAULT 0.0,
Bis DOUBLE DEFAULT 100.0,
Betrag DOUBLE DEFAULT 0.0,
Vorauskasse enum('N','Y') NOT NULL DEFAULT 'Y',
Rechnung enum('N','Y') NOT NULL DEFAULT 'Y',
Nachname enum('N','Y') NOT NULL DEFAULT 'Y',
Kreditkarte enum('N','Y') NOT NULL DEFAULT 'N',
FK_Setting_Nr INT NOT NULL DEFAULT 1,
FOREIGN KEY (FK_Setting_Nr)
          REFERENCES shop_settings (Setting_Nr),
PRIMARY KEY (Von_Bis_ID),
UNIQUE UC_Von_Bis_ID (Von_Bis_ID)
```

kreditkarte

```
Kreditkarten_ID INT DEFAULT 0 NOT NULL
            AUTO_INCREMENT,
Hersteller CHAR(32) NOT NULL,
Handling enum('intern','extern') NOT NULL
            DEFAULT 'intern',
benutzen enum('N','Y') NOT NULL DEFAULT 'Y',
PRIMARY KEY (Kreditkarten_ID),
UNIQUE UC_Kreditkarten_ID (Kreditkarten_ID)
```


© 2001 R. Glanzmann & J. Fontanil

shop_settings (Teil 1/2)

```
Setting_Nr INT NOT NULL DEFAULT 0
       AUTO_INCREMENT,
MwStsatz DOUBLE NOT NULL DEFAULT 7.6,
MwStpflichtig enum('N','Y') NOT NULL
       DEFAULT 'Y',
MwStNummer INT NOT NULL DEFAULT 0,
Name CHAR(48) NOT NULL DEFAULT ' {shop_db} ',
Adresse1 CHAR(48) NOT NULL DEFAULT 'Adresse 1',
Adresse2 CHAR(48) NOT NULL DEFAULT 'Adresse 2',
PLZOrt CHAR(48) NOT NULL DEFAULT 'PLZ und Ort',
Tel1 CHAR(24) NOT NULL
       DEFAULT 'Telefon Nummer 1',
Tel2 CHAR(24) NOT NULL
       DEFAULT 'Telefon Nummer 2',
Email VARCHAR(128) NOT NULL
       DEFAULT 'IhrShop@shopserver.com',
Admin_pwd CHAR(16) NOT NULL DEFAULT 'machshop',
Abrechnung_nach_Preis enum('N','Y') NOT NULL
       DEFAULT 'N',
Abrechnung_nach_Gewicht enum('N','Y') NOT NULL
       DEFAULT 'N',
Abrechnung_nach_Pauschale enum('N','Y') NOT NULL
       DEFAULT 'Y',
Pauschale_text varchar(127) NOT NULL
       DEFAULT 'Versand- und Verpackungskosten',
Mindermengenzuschlag enum('N','Y') NOT NULL
       DEFAULT 'N',
Mindermengenzuschlag_bis_Preis DOUBLE NOT NULL
       DEFAULT 0.0,
Mindermengenzuschlag_Aufpreis DOUBLE NOT NULL
       DEFAULT 0.0,
```

shop_settings (Teil 2/2)

```
keineVersandkostenmehr enum('N','Y') NOT NULL
       DEFAULT 'N',
keineVersandkostenmehr_ab DOUBLE NOT NULL
       DEFAULT 0.0,
Rechnung enum('N','Y') NOT NULL DEFAULT 'Y',
Nachnahme enum('N','Y') NOT NULL DEFAULT 'Y',
Vorauskasse enum('N','Y') NOT NULL DEFAULT 'Y',
Kreditkarten_Postcard enum('N','Y') NOT NULL
       DEFAULT 'N',
Nachnamebetrag DOUBLE NOT NULL DEFAULT 0.0,
Kontoinformation VARCHAR(255) NOT NULL
       DEFAULT 'Einzahlungen bitte auf unser
                Postkonto PC 40-168786-4',
Waehrung CHAR(4) NOT NULL DEFAULT 'CHF',
ShopVersion CHAR(48) NOT NULL
       DEFAULT 'Oktober 2001, DB Version v.1.05',
Gewichts_Masseinheit CHAR(16) NOT NULL
       DEFAULT 'kg',
Thumbnail_Breite INT NOT NULL DEFAULT 100,
AGB TEXT, #/* Max. 65535 Bytes gross in MySQL */
SSL enum('N','Y') NOT NULL DEFAULT 'N',
Bestellungsmanagement enum('N','Y') NOT NULL
       DEFAULT 'Y',
SuchInkrement INT NOT NULL DEFAULT 10,
max_session_time BIGINT, #/* In anderen DBs ev.
ein anderer Typ (10 Stellen) */
Opt_inc INT NOT NULL DEFAULT 3,
Var_inc INT NOT NULL DEFAULT 3,
Opt_anz INT NOT NULL DEFAULT 5,
Var_anz INT NOT NULL DEFAULT 5,
PRIMARY KEY (Setting_Nr),
UNIQUE UC_Setting_Nr (Setting_Nr)
```

backup

```
CREATE TABLE backup
(
       Backup_ID CHAR(32) NOT NULL,
       Wert CHAR(255),
       PRIMARY KEY (Backup_ID),
       UNIQUE UC_Backup_ID (Backup_ID)
);
```

2.4.6 SQL-Query Handling

Dieses Kapitel hängt eigentlich stark mit der Security (Kapitel 2.1) zusammen. Das Konzept anstatt die Query selbst in die Funktion einzubinden, dort nur Variablen zu benutzen, entstand ursprünglich in der Praktischen Arbeit dieses Frühjahrs. Dieser Ansatz hat verschiedene Vorteile: Erstens werden so die SQL-Queries von den Funktionen entkoppelt, was entscheidend ist, sobald man gedenkt eine andere Datenbank einzusetzen (und sich somit die SQL-Syntax leicht ändert). Weiter hat man so alle Queries zentral abgelegt. Bei uns in zwei Dateien und das für den ganzen Shop. Zu guter Letzt lässt sich so unsere Security gut implementieren. Wir haben dieses Konzept auch in der Diplomarbeit strikte eingehalten.

Als Anmerkung sei hier noch erwähnt, dass sich die Konvertierung der Queries am einfachsten und schnellsten mit einem Tool erledigen lässt. Wir haben hier Erstaunliches gesehen bei der Programmkombination: DATANAMICs DeZign for databases und DATANAMICs ImportER Scripts für Windows (www.datanamic.com). Diese Tools vermögen weit mehr als das einfache Konvertieren von SQLs zwischen Datenbanken-Syntax und kosten zusammen nur US$ 180.00.

 © 2001 R. Glanzmann & J. Fontanil

2.4.7 DB-Wrapper (database.php)

Von unserem Dozenten und Diplomarbeitsbetreuer, Herr Feisthammel, erhielten wir schon zu Beginn der Projektarbeit einen Datenbank Wrapper für Sybase. Dieser Wrapper stellt einerseits eine PHP Datenbank-Connection zur Verfügung, bietet die zum Datenbank-Handling nötigen Funktionen an (Query ausführen, Resultset auswerten, Resultzeilen zählen u.s.w.) und hat andererseits für jede Datenbank, auf die zugegriffen werden soll, verschiedene Klassen. Wie schon erwähnt, war dieser Wrapper für Sybase schon fertig programmiert und getestet. Während der Projektarbeit portierten wir ihn noch auf die MySQL Datenbank.

Wir benötigen die Artikel-ID eines neu eingefügten Artikels (Funktion `newArtikel` zum Einfügen des Artikels und seiner Optionen und Variationen). Dies lässt sich per MySQL sehr leicht und effizient erledigen. Sobald bei einem `INSERT` ein neuer Autowert generiert wird, kann dieser mit dem Kommando `mysql_insert_id()` ausgelesen werden. Die Exec-Funktion der TMySQLDatabase-Klasse in `database.php` liefert also nicht mehr nur true oder false zurück sondern:

- false, wenn Exec nicht korrekt ausgeführt werden konnte
- true, wenn kein Insert mit Auto-Increment Funktion ausgeführt wurde
- Den zuletzt erzeugten Auto-Increment Wert, wenn es bei einem Insert so einen gegeben hat.

Diese Erweiterung wurde bei der TSybaseDatabase-Klasse noch nicht vorgenommen. Man muss also diesen neuen Auto-Increment Wert zuerst per Select-Anweisung anfordern und dann zurückgegeben. Will man den Shop also unter Sybase laufen lassen, *muss* diese Funktion zuerst noch programmiert werden.

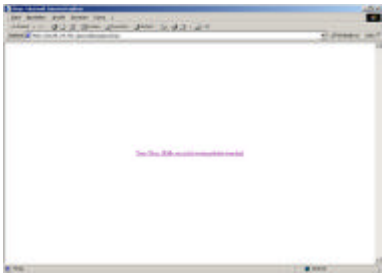
 © 2001 R. Glanzmann & J. Fontanil

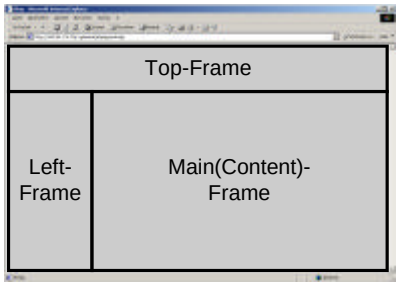
2.5 Shop Layout Management

2.5.1 Shop Layout Beschreibung

Zuerst wird die Datei `index.html` vom Webserver geladen und an den Browser ausgegeben. Diese Datei hat die Funktion, den Shopuser sofort per http-redirect auf `index.php` weiterzuleiten, welches sich im Shopverzeichnis befindet. Unterstützt der Browser diese Funktion nicht, kann sich der Surfer durch anklicken eines Links zum Shop navigieren. Ausserdem kann diese Datei noch dazu verwendet werden, Informationen für Suchrobots bereitzustellen. Wird der Shop in eine bestehende Homepage eingebettet, kann `index.html` weggelassen und der Shop direkt mit einem Link auf `index.php` aufgerufen werden.

index.html ***Frames*** ***Shop-Startseite***





Danach wird von `index.php` das Frameset aufgebaut und die Shopstartseite dargestellt. Der gesamte Shop (Benutzersicht) wird nun in diesem Dreierframeset dargestellt. Das oberste Frame (Top-Frame) wird verwendet, um Funktionen anzubieten, die jederzeit ausgeführt werden können (Warenkorb anzeigen, Zur Kasse, Hilfe, Suchfunktion) und um den Shopnamen oder das Shoplogo permanent anzuzeigen.

Im linken Frame (Left-Frame) werden die Artikelkategorien und Unterkategorien zur Auswahl angezeigt. Dieses Frame wird aktualisiert, sobald der Shopbenutzer eine Kategorie oder Unterkatgorie anklickt.

Das grösste Frame (Content-Frame) wird zur Userinteraktion verwendet. In diesem Frame läuft praktisch die gesamte Interaktion mit dem Benutzer ab.

Beim Anklicken eines Artikelthumbnails und des Hilfe-Buttons wird ein neues Fenster geöffnet. Dieses POP-UP ist mittels JavaScript realisiert. Die entsprechenden JavaScript Funktionen befinden sich direkt im HTML-Quelltext der jeweiligen Sites. Die POP-UP-Fenster können entweder per eingeblendetem Link oder mittels Schliessen-Button wieder geschlossen werden.

Es ist Shopweit ein Fontset definiert. Alle Formatierungsangaben befinden sich an zentraler Stelle in einem CSS-File, auf welches im Layout Management noch genauer eingegangen wird.

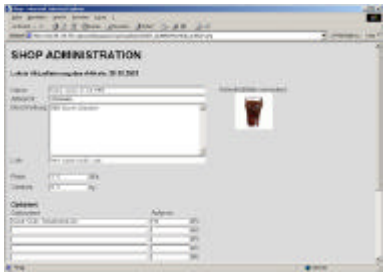
2.5.2 Layout Administrationsmodus

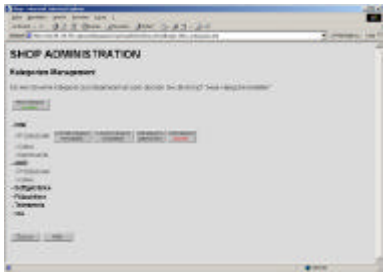
Das Benutzerinterface für den Shopadministrator überarbeiteten wir vollständig. Der Administrationsbereich besitzt nun eine eigene CSS-Datei, damit er immer gleich aussieht, auch wenn der Shopadministrator im Shoplayout-Menü Farbeinstellungen tätigt, die eine Bedienung des Shops verunmöglichen.

 © 2001 R. Glanzmann & J. Fontanil

Da im Administrationsbereich wesentlich grössere Formulare und Menüs ausgegeben werden, entschieden wir uns, auf ein Frameset zu verzichten. Alle Menüs werden ganzseitig dargestellt, wobei in der zweitobersten Zeile immer der Menütitel angezeigt wird.

Hauptmenü ***Artikel bearbeiten*** ***Kategorienmanagement***





2.5.3 Layout Management

Unser Ziel war es, einen Shop zu entwickeln, dessen optisches Erscheinungsbild beliebig auf die Bedürfnisse des Betreibers angepasst werden kann.

Um dies zu erreichen, müssen folgende HTML-Komponenten dynamisch verändert werden können:

- Hintergrundbilder/-Farben sämtlicher Frames
- Alle im Shop verwendeten Schriften-Tags
- Shoplogo, Shoptitel
- Shop-Buttons
- Frame-Grössen

Wir erstellten ein Layout-Management-Konzept, mit dem es möglich ist, alle Einstellungen des Layouts per Web vornehmen zu können. Dieses besteht aus folgenden drei Teilen:

Layout Management

Ruft der Shopadministrator das Layout Management Menu auf, werden in der Datenbanktabelle `css-file` alle Shopeinstellungen ausgelesen. Diese werden in ein HTML-Formular eingefüllt, welches dann bearbeitet werden kann. Wird dass Formular abgesendet, werden die Shopeinstellungen wieder in die Datenbank gespeichert. Zusätzlich werden die beiden Template-Files `indextemplate.txt` und `csstemplate.txt` eingelesen. Mit einem eigens zu diesem Zweck programmierten Parser werden die Files index.php und `shopstyles.css` erzeugt. Dies geschieht, indem spezielle Tags, welche sich in eckigen Doppelklammern befinden (`<<Tagname>>`), durch entsprechende Shopeinstellung aus der Datenbank ersetzt werden:

Zeile im Template-File:

```
body.left { color:<<left_font_c>>; background:<<left_bg_c>>; <<left
_bg_img>> }
```

Werte in der Datenbank:

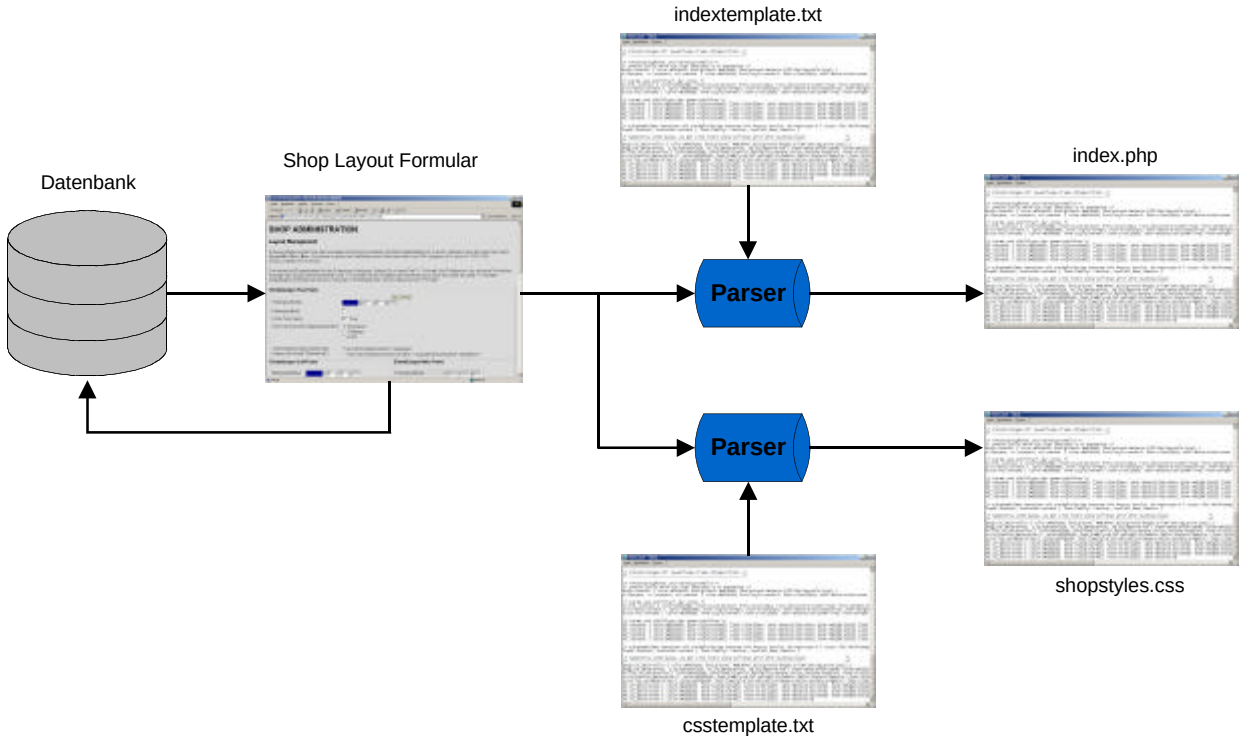
```
left_font_c = #ffffff
left_bg_c = #00008b
left_bg_img = background-image:url(Bilder/bg_left.jpg);
```

Zeile, wie sie in die Datei `shopstyles.css` eingefügt wird:

 © 2001 R. Glanzmann & J. Fontanil

```
body.left { color:#ffffff; background:#00008b; background-image:
url(Bilder/bg_left.jpg); }
```

Ablauf des Layout Management Vorgangs:



Hintergrundbilder und Shoplogo hochladen

Die Hintergrundbilder und das Shoplogo können mittels Web-Formular hochgeladen werden. Falls der Upload erfolgreich ausgeführt werden konnte, wird das entsprechende Bild im Verzeichnis `shop/Bilder` überschrieben. Da es bei eingeschaltetem SafeMode nicht möglich ist, neue Files im Filesystem zu erzeugen, werden bei der Installation des Shops einige Dummy-Dateien erstellt. Bereits existierende Dateien können auch bei eingeschaltetem SafeMode überschrieben werden.

Die akzeptierten Bildtypen zum Upload sind: GIF, JPEG und PNG. Anhand der Dateiendung wird schon vor dem hochladen mittels JavaScript überprüft, ob es sich um einen akzeptierten Bildtyp handelt.

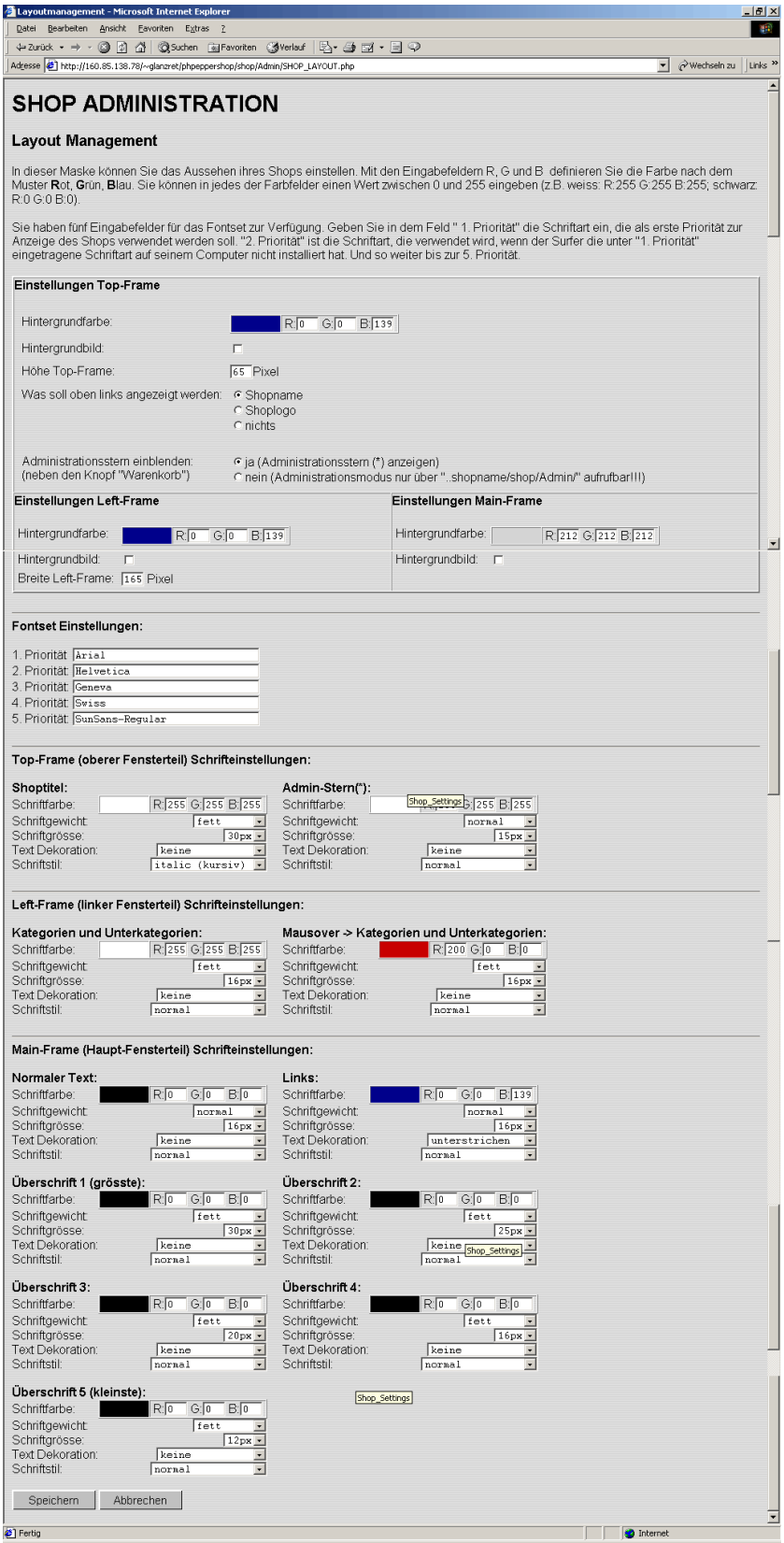
Da man im Layout Management wählen kann, ob ein Hintergrundbild und Shoplogo verwendet werden soll, muss nach erfolgreichem hochladen eines Bildes, die Datenbank aktualisiert werden. Eine Aktualisierung der Datenbank zieht automatisch auch eine Neuerzeugung des CSS-Files nach sich.

Buttons hochladen

Nach dem gleichen Verfahren, mit dem die Hintergrundbilder hochgeladen werden, können auch die Shopbuttons durch eigene ersetzt werden. Die Buttons müssen vom Typ GIF sein. Hätten wir auch andere Bildtypen zugelassen, wäre das eine ernsthafte Performancegefahr für den Shop gewesen, da der Typ jedes ausgegebenen Buttons in der Datenbank hätte ermittelt werden müssen. Es wären dadurch eine Unmenge von SQL-Abfragen entstanden, die wir unbedingt vermeiden wollten.

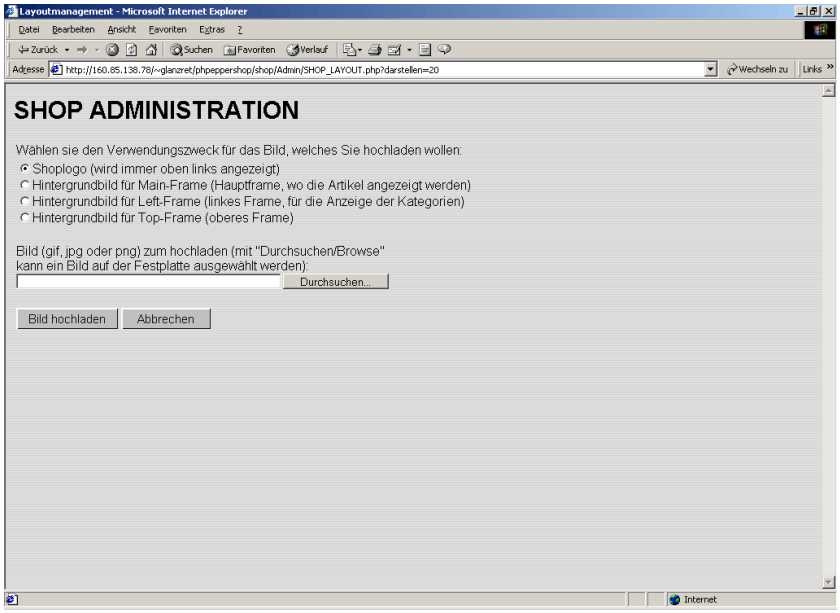
 © 2001 R. Glanzmann & J. Fontanil

Screenshot des Layout Managements:

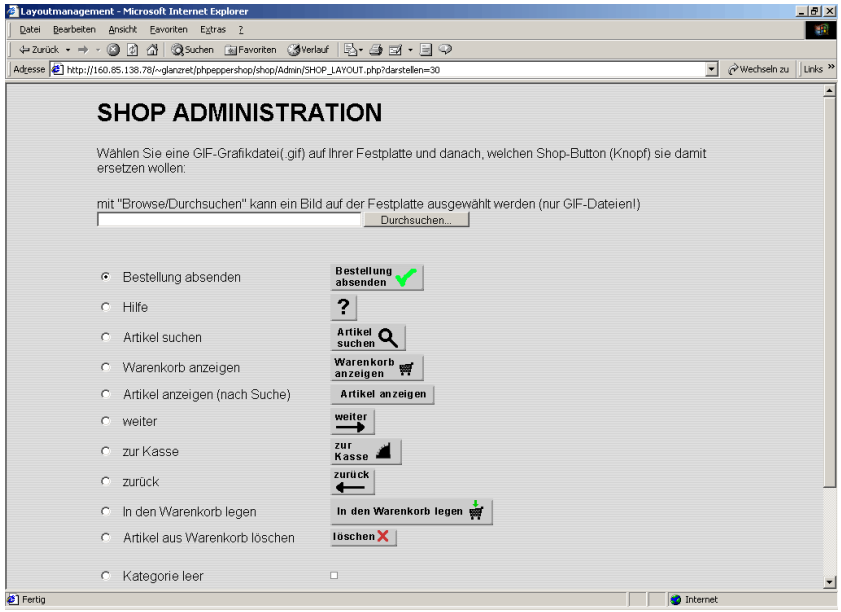


 © 2001 R. Glanzmann & J. Fontanil

Screenshot Bilder hochladen:


Layoutmanagement - Microsoft Internet Explorer
SHOP ADMINISTRATION
Wählen sie den Verwendungszweck für das Bild, welches Sie hochladen wollen:
Shoplogo (wird immer oben links angezeigt)
Hintergrundbild für Main-Frame (Hauptframe, wo die Artikel angezeigt werden)
Hintergrundbild für Left-Frame (linkes Frame, für die Anzeige der Kategorien)
Hintergrundbild für Top-Frame (oberes Frame)
Bild (gif, jpg oder png) zum hochladen (mit "Durchsuchen/Browse" kann ein Bild auf der Festplatte ausgewählt werden):
Durchsuchen...
Bild hochladen
Abbrechen

Screenshot Buttons hochladen:



2.5.4 Buttons im Administrationsbereich

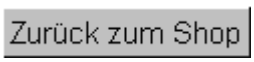
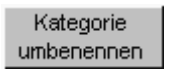
Da nach Abschluss der Diplomarbeit prinzipiell jedermann am PhPepperShop weiterentwickeln kann, werden neue Menüs im Administrationsbereich dazu kommen. Damit trotzdem eine optische Einheitlichkeit bestehen bleibt, haben wir genau definiert, wie Administrations-Buttons auszusehen haben.

Wir verwendeten das Freeware Programm „Buttonz“ von Björn Ischo zur Erzeugung der Buttons. Da die Homepage von Björn Ischo russisch ist, empfehlen wir, das Programm von http://www.pctip.ch herunterzuladen.

Die Templates zur Erzeugung der Buttons befinden sich auf der Diplomarbeit-CD. Sie können

 © 2001 R. Glanzmann & J. Fontanil

auch direkt bei uns (developers@phpepeppeshop.com) bezogen werden.

Template-File	*Abmessungen (b*h)*	*Beispiel-Button*
`bt_template_einzeilig_82px.bff`	82px * 26px	
`bt_template_einzeilig_110px.bff`	110px * 26px	
`bt_template_einzeilig_125px.bff`	125px * 26px	
`bt_template_zweizeilig.bff`	81px * 32px	

Folgende Namenskonvention sollte beim Erstellen neuer Buttons für den Administrationsbereich eingehalten werden:

`bt_ + „Buttonbeschriftung“ + _admin.gif`

Wird zum Beispiel ein Button mit der Beschriftung „Backup“ eingeführt, heisst die Datei folgendermassen: `bt_backup_admin.gif`

2.5.5 Ändern der Welcome Page

Um die Willkommens-Page abzuändern, muss man nur die Datei `content.html` im Verzeichnis: `<shopdir>/shop/Frameset/` ändern oder ersetzen.

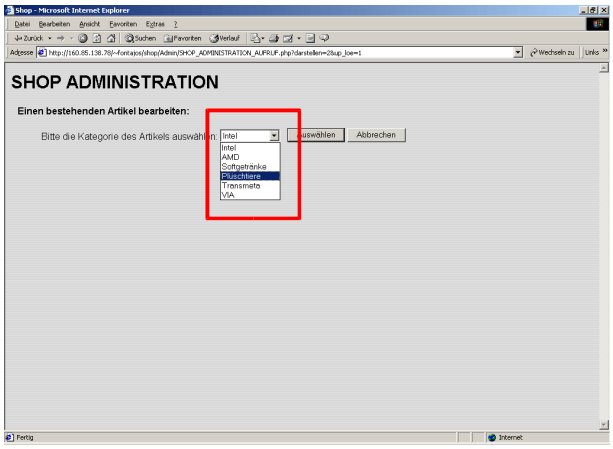
 © 2001 R. Glanzmann & J. Fontanil

2.6 Artikelmanagement

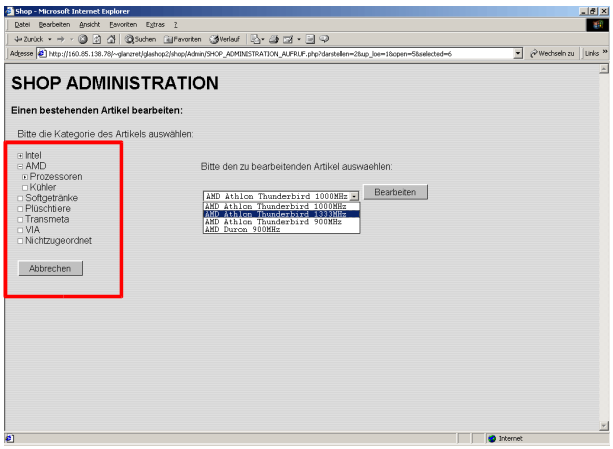
Mit der Einführung von Unterkategorien und der Möglichkeit, Artikel mehreren Kategorien zuordnen zu können, waren wir gezwungen, auch das Artikelmanagement gründlich zu überarbeiten. Wir nutzten diesen Umstand, um gleich auch noch die Eingabeüberprüfung mittels PHP durch JavaScript Funktionen zu ersetzen.

Da unser ursprüngliches Menü, zur Auswahl eines Artikels für die Bearbeitung, nicht Unterkategorien tauglich war, mussten wir uns eine andere Lösung zur Auswahl der Kategorie/Unterkategorie des Artikels ausdenken. Das im Left-Frame des Shops verwendete Kategorien-Menü konnte nach einigen Modifikationen sehr gut für diesen Zweck eingesetzt werden. Zur Auswahl des Artikels innerhalb der Kategorie blieben wir bei der bereits in der ersten Version des Shops verwendeten Pull-Down Liste.

Alte Version zur Kategorieauswahl



Neue Version zur Kategorieauswahl

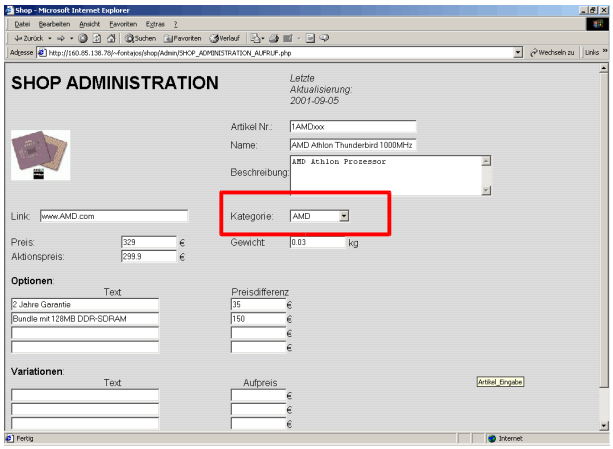


An der Bearbeitung der Artikeldaten änderte sich eigentlich nichts. Da wir die Anzahl Optionen- und Varianten-Felder neu dynamisch konfigurierbar implementierten, mussten wir noch einige Änderungen an der Darstellung des Formulars vornehmen. Die Zuordnung des Artikels zu einer Kategorie konnte nicht mehr im gleichen Formular vorgenommen werden, da der Artikel mit dem neuen Datenbankmodell auch mehreren oder keiner Kategorie zugeordnet werden kann.

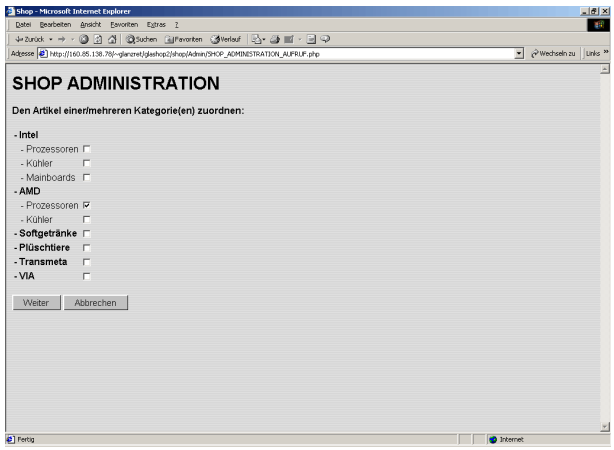
Wir mussten also einen weiteren Zwischenschritt einfügen, in dem der Artikel beliebig vielen Kategorien zugeordnet werden konnte. Zu diesem Zweck geben wir die Kategorienliste mit allen Kategorien aufgeklappt aus. Hinter jeder Kategorie und Unterkategorie wird eine Checkbox angezeigt. Soll der Artikel einer Kategorie zugeordnet werden, wird einfach die entsprechende Checkbox angewählt. Enthält eine Kategorie Unterkategorien, können der Kategorie selbst keine Artikel zugeordnet werden.

 © 2001 R. Glanzmann & J. Fontanil

Alte Version Kategorienzuordnung

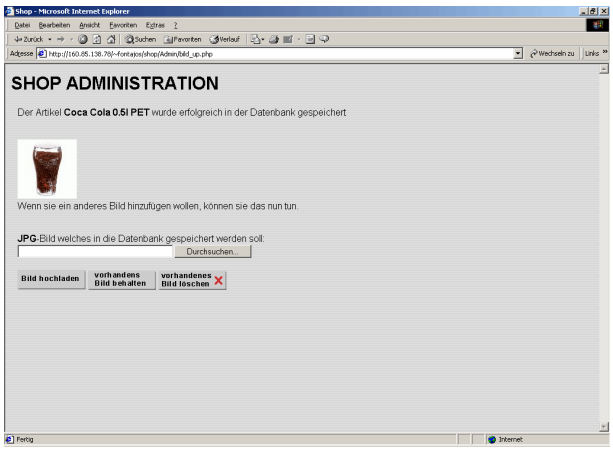


Neue Version Kategorienzuordnung

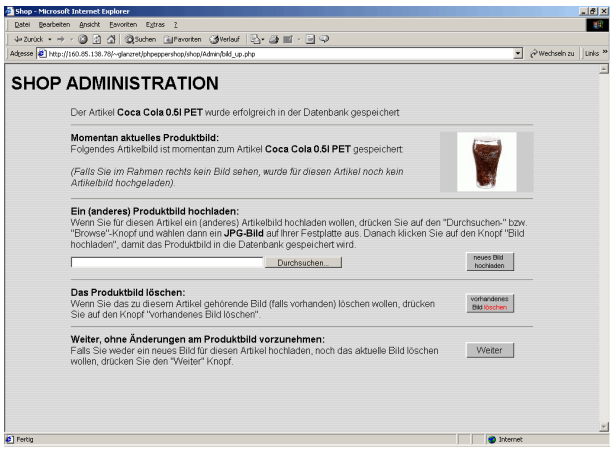


Als letzter Schritt kann für einen Artikel noch ein Bild in die Datenbank gespeichert werden. Der Shopadministrator hat die Wahl, ob er das aktuelle Produktbild löschen, dem Artikel eine neues hinzufügen oder keine Änderungen am Artikel vornehmen will. Das von uns in der Projektarbeit erstellte Formular war aber mehr verwirrend, als hilfreich. Wir wollten, damit die Artikelbearbeitung weiterhin effektiv ausgeführt werden kann, keinen weiteren Zwischenschritt einführen. Deshalb entschlossen wir uns, aussagekräftige Buttons zu erstellen, Hilfetexte direkt in die HTML-Seite auszugeben und die Seite mit horizontalen Querbalken optisch in drei Teile zu gliedern.

Alte Version des Bildmenüs


SHOP ADMINISTRATION
Der Artikel Coca Cola 0.5l PET wurde erfolgreich in der Datenbank gespeichert
Wenn sie ein anderes Bild hinzufügen wollen, können sie das nun tun.
JPG-Bild welches in die Datenbank gespeichert werden soll:
Durchsuchen...
Bild hochladen
vorhandenes Bild behalten
vorhandenes Bild löschen

Neue Version des Bildmenüs


SHOP ADMINISTRATION
Der Artikel Coca Cola 0.5l PET wurde erfolgreich in der Datenbank gespeichert
Momentan aktuelles Produktbild:
Folgendes Artikelbild ist momentan zum Artikel Coca Cola 0.5l PET gespeichert:
(Falls Sie im Rahmen rechts kein Bild sehen, wurde für diesen Artikel noch kein Artikelbild hochgeladen).
Ein (anderes) Produktbild hochladen:
Wenn Sie für diesen Artikel ein (anderes) Artikelbild hochladen wollen, drücken Sie auf den "Durchsuchen-" bzw. "Browse"-Knopf und wählen dann ein JPG-Bild auf Ihrer Festplatte aus. Danach klicken Sie auf den Knopf "Bild hochladen", damit das Produktbild in die Datenbank gespeichert wird.
Durchsuchen...
neues Bild hochladen
Das Produktbild löschen:
Wenn Sie das zu diesem Artikel gehörende Bild (falls vorhanden) löschen wollen, drücken Sie auf den Knopf "vorhandenes Bild löschen".
vorhandenes Bild löschen
Weiter, ohne Änderungen am Produktbild vorzunehmen:
Falls Sie weder ein neues Bild für diesen Artikel hochladen, noch das aktuelle Bild löschen wollen, drücken Sie den "Weiter" Knopf.
Weiter

Eine weitere Neuerung ist die Kategorie „Nichtzugeordnet“. Dieser Kategorie enthält alle Artikel, die keiner Kategorie zugeordnet sind. Sie wird im Shop nirgends angezeigt und dient lediglich als Containment für Artikel, die momentan im Shop nicht verwendet werden, oder deren Kategorie bzw. Unterkategorie gelöscht wurde.

Erstellt der Shopadministrator einen neuen Artikel, wird der gleiche Ablauf verwendet, wie bei der Mutation. Per JavaScript wird überprüft, ob die minimal erforderlichen Angaben eingegeben wurden.

An der Funktion für das Löschen eines Artikels aus dem Shop wurde praktisch nichts verändert.

 © 2001 R. Glanzmann & J. Fontanil

2.6.1 Artikelsuche

Die Artikelsuche wurde in der Diplomarbeit stark erweitert.

Mehrere Suchbegriffe in einer Abfrage erlaubt

Im Hinblick auf Warensortimente mit vielen ähnlichen Produkten, haben wir die Unterstützung von mehreren Suchbegriffen in einem Suchstring eingeführt. Die Wörter werden, wie von der Suchmaschine Google® her gewohnt, AND-verknüpft.

Bilder können optional angezeigt werden

Seitdem wir die SQLs und die Auslesen-Schleife in getgesuchterArtikel(..) überarbeitet haben, werden nicht mehr nur der Artikelname und dessen Artikel-ID zurückgegeben, sondern auch gleich noch weitere Attribute. Mit dieser Vorarbeit war es ein Leichtes, die Bilder optional als Suchergebnis anzeigen zu lassen. Damit die Bedienung intuitiv bleibt, wurde das jeweilige Artikelbild auch mit dem Link auf den Artikel versehen.

Mehrfachkategorien Anzeige

Seitdem der PhPepperShop Artikel in mehreren Kategorien gleichzeitig abgelegt werden kann, ist es auch wichtig, in der Suchfunktion des Shops darauf hinzuweisen, wenn ein Artikel in mehreren Kategorien vorhanden ist.

Wir haben uns entschieden nicht nur auf das mehrfache Vorhandensein hinzuweisen, sondern die Mehrfachkategorie-Anzeige gleich mit entsprechenden Links auszustatten. Wir haben dafür die Klasse Artikelmitkategorien (artikel_def.php) eingeführt und die Funktion getgesuchterArtikel(..) entsprechend erweitert. Auf diese Weise kann der Shopbenutzer bequem in die Kategorie springen, in der er den Artikel erwartet hätte.

Beispielsanzeige:

Dieser Artikel ist in mehreren Kategorien vorhanden: [Intel -> Prozessoren] [AMD -> Prozessoren]

Inkrementangabe

Eine Suche kann sehr aufwändig sein. Es wird eine Query mit mehreren konjunktiv verknüpften LIKE Bedingungen auf die Attribute Name und Beschreibung *aller* Artikel in der Datenbank ausgeführt. Wenn man die Suchbegriffe nicht geschickt auswählt, kann es gut sein, dass man eine enorm grosse Zahl an Treffern erhält. Dies belastet einerseits die Datenbank, sie muss alle Daten auslesen und andererseits die Kundenverbindung, weil alle Daten bis zum Shopbenutzer übertragen werden müssen. Um diesem Problem Herr zu werden, haben wir die Anzahl gleichzeitig angezeigter Suchresultate auf eine in den „allgemeinen Shopeinstellungen“ konfigurierbare Anzahl beschränkt.

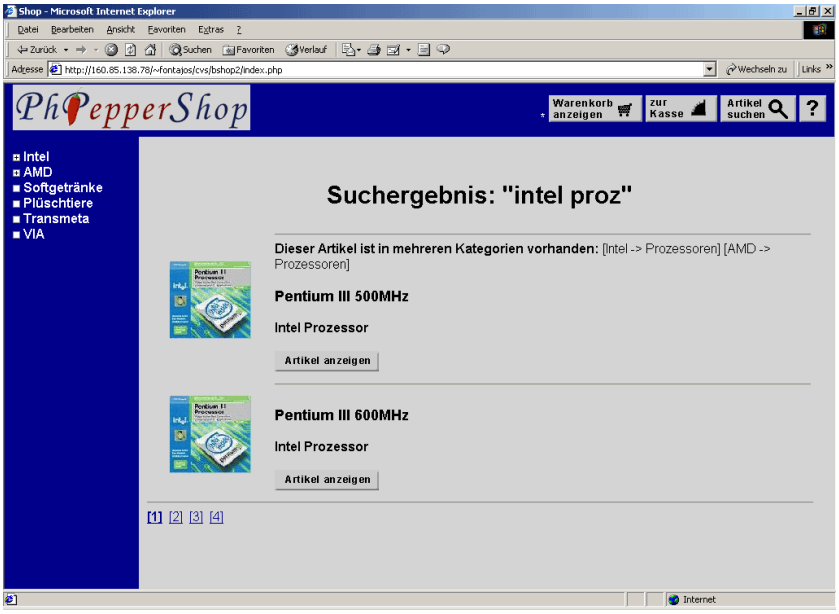
Der Funktion getgesuchterArtikel(..) wird ein Startwert und eine Anzahl übergeben. Bei der ersten Anzeige ist der Startwert auf 0 gesetzt. Dem Shopbenutzer wird, sobald es mehr Treffer gibt als die konfigurierte maximale Anzahl gleichzeitig angezeigter Artikel, ein Auswahlmenü eingeblendet. In diesem Menü kann er zwischen den Resultat-Intervallen hin und her springen. Beispiel: **[1]** [2] [3] [4]

Realisiert haben wir diese Funktion mit der LIMIT Angabe. Leider wussten wir nicht, dass LIMIT ein MySQL-Spezifischer Befehl ist und nicht zum SQL'92 Standard gehört. Diese Funktion müsste also noch umprogrammiert werden, sobald man eine andere Datenbank als MySQL einsetzen will.

 © 2001 R. Glanzmann & J. Fontanil

Screenshot

Hier sieht man ein Suchresultat mit aktivierter Bildanzeige, einem Artikel, welcher in zwei verschiedenen Kategorien eingeteilt ist und einer maximal gleichzeitig angezeigten Anzahl Artikel von zwei:



 © 2001 R. Glanzmann & J. Fontanil

2.7 Kategorienmanagement

Das Kategorienmanagement ist sehr umfangreich aufgebaut. Kategorien und Unterkategorien können erstellt, umbenennt, gelöscht oder an eine beliebige Stelle verschoben werden. Entfernt man eine Kategorie, kann man wählen, ob die in ihr enthaltenen Artikel mitgelöscht oder der Kategorie „Nichtzugeordnet“ zugewiesen werden sollen (sofern sie nicht noch einer anderen Kategorie zugeordnet sind).

Wir erstellten gleich zu Beginn der Implementierung ein detailliertes Konzept, welches den Funktionsumfang und Ablauf, sowie die HTML-Ausgabe genau beschreibt. An dieses hielten wir uns konsequent. Da daraus sehr gut die Abläufe zu erkennen sind, wird es hier in *unveränderter Form* abgedruckt.

Im Administrationstool wurden weiter einige Buttons und Texte wurden zwecks besserer Verständlichkeit nochmals überarbeitet (siehe Screenshots weiter unten). Zusätzlich kann beim Löschen einer Kategorie bzw. Unterkategorie noch gewählt werden, ob die in ihr enthaltenen Artikel unwiderruflich gelöscht werden.

2.7.1 Ursprüngliches Konzept (unverändert)

Es gibt Kategorien und Unterkategorien. Eine Kategorie kann beliebig viele Unterkategorien enthalten.

Funktionen, die mit Kategorien und Unterkategorien ausgeführt werden können:

Kategorie:	- neue Kategorie erstellen	[1]
	- Kategorie verschieben	[2]
	- Kategorie umbenennen	[3]
	- Kategorie löschen	[4]
Unterkategorien:	- neue Unterkategorie erstellen	[5]
	- Unterkategorie innerhalb Kategorie verschieben	[6]
	- Unterkategorie in eine andere Kategorie verschieben	[7]
	- Unterkategorie umbenennen	[8]
	- Unterkategorie löschen	[9]

Kategorienmanagement Bildschirmansicht:

[1]

Prozessoren	[5]	[4]	[2]	[3]
Intel	[6]	[7]	[8]	[9]
AMD	[6]	[7]	[8]	[9]
Motorola	[6]	[7]	[8]	[9]
Gehäuse	[5]	[4]	[2]	[3]
Catan	[6]	[7]	[8]	[9]
Evercase	[6]	[7]	[8]	[9]
Monitore	[5]	[4]	[2]	[3]
Drucker	[5]	[4]	[2]	[3]

 © 2001 R. Glanzmann & J. Fontanil

neue Kategorie erstellen [1]:

Kategoriename: [ ]

Wo soll die Kategorie einfügt werden:

hier einfügen
Prozessoren
hier einfügen
Gehäuse
hier einfügen
Monitore
hier einfügen
Drucker
hier einfügen

abbrechen

Kategorie verschieben [2]:

Wohin soll die Kategorie Gehäuse verschoben werden:

hierher verschieben
Prozessoren
hierher verschieben
Monitore
hierher verschieben
Drucker
hierher verschieben

abbrechen

Kategorie umbenennen [3]:

Neuer Kategoriename für die Kategorie Gehäuse: [ ]

ok abbrechen

Kategorie löschen [4]:

Wenn Sie die Kategorie löschen, werden alle Unterkategorien, die sich darin befinden gelöscht. Artikel, die sich allenfalls noch in dieser Kategorie oder einer ihrer Unterkategorien befinden, werden der Kategorie 'nicht zugeordnete Artikel' zugewiesen.

Wollen Sie die Kategorie Gehäuse mit allen Unterkategorien löschen?

ja nein

neue Unterkategorie erstellen [5]:

Unterkategoriename: [ ]

Wo soll die Unterkategorie innerhalb der Kategorie Gehäuse einfügt werden:

hier einfügen
Catan
hier einfügen
Evercase

 © 2001 R. Glanzmann & J. Fontanil

hier einfügen

abbrechen

Unterkategorie innterhalb Kategorie verschieben [6]:

Wohin soll die Unterkategorie AMD verschoben werden:

hierher verschieben
Intel
hierher verschieben
Motorola
hierher verschieben

abbrechen

Unterkategorie in eine andere Kategorie verschieben [7]:

Wohin soll die Unterkategorie AMD verschoben werden:

Prozessoren
- hierher verschieben
- Intel
- hierher verschieben
- AMD
- hierher verschieben
- Motorola
- hierher verschieben

Gehäuse
- hierher verschieben
- Catan
- hierher verschieben
- Evercase
- hierher verschieben

Monitore
- hierher verschieben

Drucker
- hierher verschieben

abbrechen

Unterkategorie umbenennen [8]:

Neuer Unterkategoriename für die Unterkategorie Catan:

[ ]

ok abbrechen

Unterkategorie löschen [9]:

Wenn Sie die Unterkategorie löschen, werden alle Artikel, die sich noch in der Unterkategorie befinden der Kategorie 'nicht zugeordnete Artikel' zugewiesen.

Wollen Sie die Unterkategorie Catan löschen?

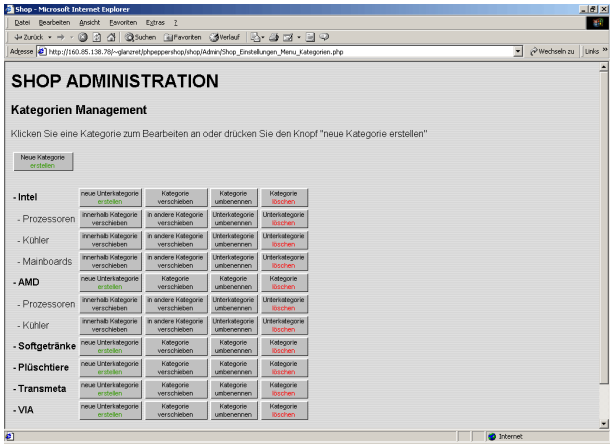
ja nein

 © 2001 R. Glanzmann & J. Fontanil

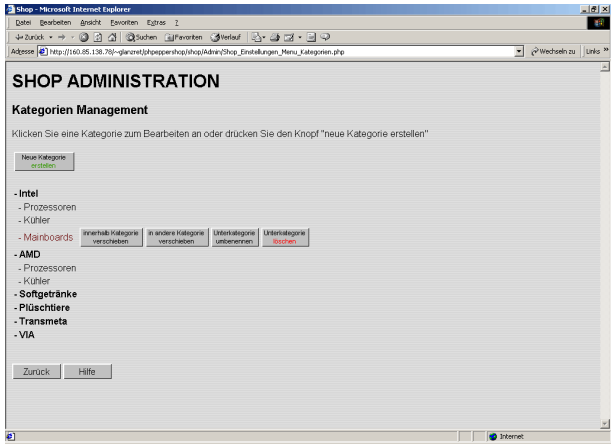
2.7.2 Hauptmenü

Zu Beginn war geplant, das Hauptmenü des Kategorienmanagements folgendermassen auszugeben: Kategorienliste aufgeklappt mit allen Unterkategorien, rechts davon jeweils die zugehörigen Buttons. Wie im folgenden Bild zu erkennen ist, wurde durch die gegebene Buttonhöhe die Kategorienliste auseinandergezogen. Ausserdem war das Menü durch die grosse Anzahl der Buttons kaum mehr bedienbar.

schlecht bedienbares Kategorienmanagement!



gut bedienbares Kategorienmanagement!



JavaScript bietet glücklicherweise die Möglichkeit, Bilder auf einer ausgegebenen HTML-Seite dynamisch zu ersetzen. Wir entschieden uns dann zuerst für eine sehr dynamische Lösung. Sobald man mit er Maus auf eine Kategorie fährt, sollen gleich neben der Kategorie die Auswahlmöglichkeiten eingeblendet werden. Diese Idee scheiterte leider an der langsamen JavaScript-Engine der Netscape und Mozilla-Browser.

Durch eine kleine Abänderung des JavaScripts von onMouseOver (Aktion ausführen, wenn die Maus über ein Objekt fährt) auf onMouseClick (Aktion ausführen, wenn man mit der Maus auf ein Objekt klickt), entstand dann das schlussendlich verwendete, mit allen Browsern bedienbare Menü. Wird ein Kategorie-, bzw. Unterkategoriename angeklickt, werden rechts davon die für diese Kategorie verfügbaren Auswahlmöglichkeiten eingeblendet. Diese bleiben solange stehen, bis eine andere Kategorie angeklickt wird.

Leider kann mit Netscape 4.7x die Grösse eines einmal ausgegebenen Bildes nicht mehr dynamisch verändert werden. Aus diesem Grund ist eine Browserweiche implementiert, die für Netscape-Browser jeden Button schon bei der Ausgabe des HTML-Dokuments in der vollen Grösse, transparent ausgibt. Da sich dadurch der Abstand der Kategorienamen erhöht, wird Netscape 4.7 Benutzern folgender Hinweis ausgegeben:

> D*ieser Shopadministrationsmodus ist nicht für Browser der Generation Netscape 4.xx optimiert. Es sind alle Funktionen verfügbar. Die Darstellung ist jedoch nicht optimal!*

 © 2001 R. Glanzmann & J. Fontanil

2.8 Zahlungsmöglichkeiten

2.8.1 Vorauskasse, Rechnung, Nachnahme und kostenlos

Bei vielen kleineren Shops kommen teure Mailorder Verträge nicht in Frage. Für sie sind die üblichen Zahlungsmethoden wie Vorauskasse, Rechnung und Nachnahme die erste Wahl. Mehrere Leute haben uns empfohlen, Vorauskasse als Zahlungsmethode zu implementieren. Wir nahmen uns das zu Herzen. Bei Vorauskasse ist es praktisch, wenn man die Kontoinformationen gleich mit angeben kann.

Bei der Nachnahmezahlung haben wir noch eine in den Versandkosten getrennt einstellbare Nachnahmegebühr eingeführt. Man kann die Kosten auf diese Weise für den Kunden transparent aufstellen.

Was, wenn ein Kunde nur einen Gratiskatalog bestellt? Wir haben für Bestellungen deren Artikeltotal CHF 0.00 beträgt eine Spezialbehandlung implementiert. Es werden dann keinerlei Versandkosten und Mindermengenzuschläge berechnet.

2.8.2 Kreditkarten

Wir haben Kreditkarten als Zahlungsmittel implementiert, weil sie allen Unkenrufen des Datenschutzes wegen zum Trotz, eine oft anerkannte und weit verbreitete Zahlungsart darstellen. Weiter kann der Shopadministrator vor der Auslieferung der Bestellung an den Kunden bei den Kreditkartenanbietern verifizieren, ob er sein Geld auch erhalten wird. Diese Garantie hat er bei einer Zahlung auf Rechnung nicht.

Shop-intern abgewickelte Kreditkartenzahlung

Bei einer Shop-internen Kreditkartenzahlung werden die Kreditkartendaten im PhPepperShop erfasst und per E-Mail an den Shopadministrator gesendet. Wir haben diesen Weg der Kreditkartenzahlung implementiert, damit man das günstigst mögliche Kreditkartenhandling benutzen kann: Es wird *nur* ein Mailorder Vertrag mit dem entsprechenden Kreditkartenanbieter benötigt, mehr nicht. Es entstehen keine weiteren Kosten für Payment Institute wie z.B. yellowpay oder Saferpay. Auf diese Weise können es sich auch kleinere Shops leisten Kreditkarten zu unterstützen.

Einschränkungen

Wir wissen bis jetzt *nur* von Eurocard/Mastercard und VISA, dass sie einen Mailorder Vertrag mit einem Shopbesitzer abschliessen, sobald dieser beweisen kann, dass die Kreditkartendaten nur verschlüsselt übertragen werden. Dies bedingt, dass die Kreditkartendaten bei der Eingabe vom Kunden per SSL verschlüsselt übertragen werden. Weiter muss das E-Mail an den Shopadministrator verschlüsselt werden (z.B. per GNU-PG).

PhPepperShop bietet die (optionale) SSL-Verschlüsselung. Auch wurde der Bestellablauf soweit fertig gestellt, dass man nur noch den E-Mail String verschlüsseln muss (Modul: USER_BESTELLUNG_1.php, Teil: darstellen = 4). Leider konnten wir die Schwierigkeiten die uns PHP's SAFE-MODE bereitete nicht mehr rechtzeitig auf Diplomarbeitsende lösen, so dass dieses Proof of Concept noch nicht integriert wurde. Wir reichen diese Erweiterung aber baldmöglichst nach.

Shop-extern abgewickelte Kreditkartenzahlung

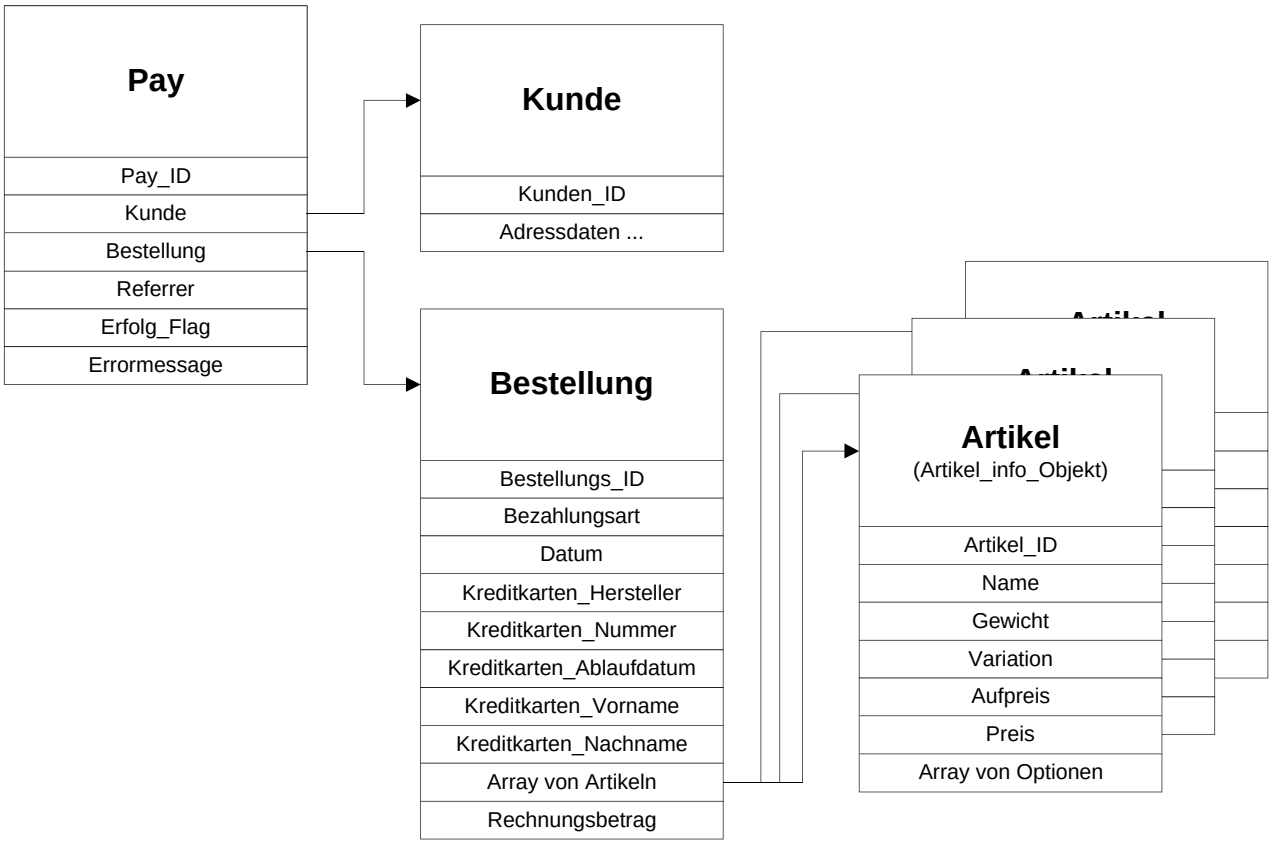
Damit jemand auch die Vorteile eines professionellen Payment Instituts benutzen kann, ha-

 © 2001 R. Glanzmann & J. Fontanil

ben wir eine Zahlungsschnittstelle eingebaut. Hier kann eine Bestellung an ein externes Zahlungssystem weiter gereicht werden, welches die Zahlung übernimmt. Das externe Zahlungssystem liefert danach eine Meldung über den Erfolg der Zahlung und leitet den Shopbenutzer wieder zu einer vordefinierten Seite im Bestellungsablauf des PhPepperShop zurück. Ein Ablaufdiagramm des Bestellungsvorgangs kann im Kapitel 1.7.3 (Tests) angeschaut werden.

Idee und Konzept

Die Zahlungsschnittstelle (payment_interface.php) ist eine sehr wichtige Schnittstelle. Man kann mit ihr eine komplette Bestellung bis ins letzte Detail ansehen:



Usability der Schnittstelle

Wir haben einigen Aufwand betrieben um zu vermeiden, dass sich der Benutzer der Schnittstelle mit dem Code des PhPepperShop befassen muss. Dies dient auch der Sicherheit. Es ist immer ein Risiko, wenn jemand am Code des eigentlichen Programms Veränderungen vornehmen muss.

Die Zahlungsschnittstelle befindet sich in der Datei `payment_interface.php`. Wir haben dort ziemlich detailliert aufgeschrieben, wie man dieses Modul benutzt:

Das Payment Interface definiert ein Formular, welches einem externen Zahlungssystem über hidden-Fields die benötigten Daten zur Verfügung stellt. In der Variable $pay_here kann man angeben, wo das externe Zahlungssystem die Daten übernimmt.

Das Modul wurde in zwei Teile gegliedert. Im ersten Teil, kann man seine Angaben machen, URL u.s.w., im zweiten Teil (HTML-Formular) muss man nur etwas ändern, wenn man weitere Attribute übergeben will. Wir übertragen per Default alle Kundenattribute und alles was

 © 2001 R. Glanzmann & J. Fontanil

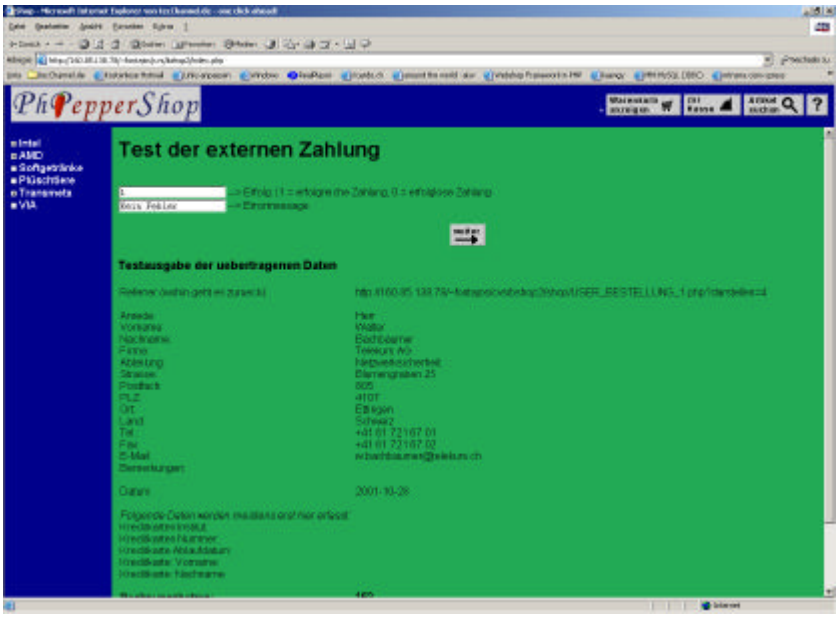
die Bestellung hat. Lediglich Artikeldetails werden nicht automatisch übertragen. Wir glauben deshalb, dass es nicht so schnell nötig sein wird, dieses Formular zu erweitern. Falls dies trotzdem gebraucht wird, kann man in der Datei `payment_interface.php` nachlesen wie man vorgehen muss.

Die einzige Funktion im Modul: `payment_extern($Pay)` mit einem Pay-Objekt als Argument, wird vom Shop aus aufgerufen (Modul: USER_BESTELLUNG_1.php,darstellen=6). Sie stellt einen Formularteil zusammen, welcher wie in einer Darstellen-Funktion (siehe Schichten des PhPepperShops, Kapitel 2.2.1) direkt als HTML zurück geschrieben wird.

Ist die Zahlung abgeschlossen, so soll vom externen Zahlungssystem auf die Seite `$Referrer` zurückgesprungen werden. Im Flag `$Erfolg` kann man mit 1 Erfolg und mit 0 keinen Erfolg der externen Zahlung signalisieren. Im Fehlerfall kann man dem PhPepperShop in der Variable `$Errormessage` einen String mit einem genaueren Fehlerbeschrieb mitgeben.

Das Testmodul (pay_ext_test.php)

Wenn man den Shop neu installiert und eine Bestellung mit einer Kreditkarte mit externem Handling tätigt, so wird vom `payment_interface.php` standardmässig die 'Emulation einer externen Zahlungsstelle' (`pay_ext_test.php`) aufgerufen. Sie zeigt alle übergebenen Variablen an und man kann zwischen Erfolg und Misserfolg der Zahlung wählen. Man kann hier optional eine Errormessage mitgeben. Dieses Modul dient als Test, falls jemand diese Schnittstelle erweitern möchte. Hier ein Screenshot, nachdem man den AGB akzeptiert hat:



 © 2001 R. Glanzmann & J. Fontanil

2.9 Bestellungsmanagement

Unter einem Bestellungsmanagement verstanden wir die Möglichkeit, offene Bestellungen abzurufen, Bestellungen zu löschen und Kundenadressen (Lieferadressen) zu mutieren - eigentlich eine Verwaltung von offenen Bestellungen.

Wir mussten uns schon früh Gedanken über das Bestellungsmanagement machen, angefangen beim Session Management, bei welchem nach Bestellungsabschluss eines Kunden alle nicht abgeschlossenen, abgelaufenen Sessions gelöscht werden. Hier muss schon überprüft werden, ob auch alle abgeschlossenen Bestellungen von temporären Kunden gelöscht werden. Weiter muss der Shopadministrator das Bestellungsmanagement ein- oder ausschalten können. Es gibt sicher viele kleinere Shops, bei welchen der Shopadministrator die ganze Bestellungsverwaltung über die E-Mails abwickelt.

Da wir auch Manipulationen erlauben wollen (löschen, mutieren), muss die referentielle Integrität von Kunde und Bestellung logisch implementiert werden, denn die Datenbank kann nicht zwei verschiedene Abhängigkeiten mit nur zwei Tabellen nachbilden. Wir haben das auf der Funktionsebene gelöst, indem wir Funktionen zur Kundenbearbeitung, weitere Funktionen zur Bestellungsbearbeitung und wieder andere zum Verändern von Beziehungen zwischen Kunde und Bestellung erstellt haben.

Funktional ist das Bestellungsmanagement fertig ausprogrammiert. Einzig die HTML-Masken zum Bearbeiten der Bestellungen erlauben momentan nur die gleichzeitige Darstellung einer Bestelllung mit ihrem Kundendatensatz. Bestellungen können bis jetzt nur aufgrund ihrer Referenznummer angezeigt werden. Dies ist für den Fall gedacht, wenn ein Shopbenutzer nach getätigter Bestellung Änderungen wünscht. Was sicher nächstens implementiert wird, ist die Suche nach Kundennamen und allen abgeschlossenen, noch nicht gelöschten Bestellungen.

 © 2001 R. Glanzmann & J. Fontanil

2.10 Kundenattribute Management

Dank dem Kundenattribute Management ist es dem Administrator möglich zu bestimmen, welche Adressinformationen ein Shopkunde eingeben kann und welche von diesen er eingeben muss.

Die Kundenattribut-Bezeichnungen sind in der Tabelle „`attribut`“ gespeichert. Alle Attribut-Bezeichnungen sind in der Datenbank gespeichert und könnten theoretisch dynamisch verändert werden. Wir haben ein Front-End erstellt, welches 14 fest definierte und vier frei konfigurierbare Attribute enthält. Die fest definierten Attribute sind die Kundenstammdaten.

Die vier Zusatzattribute können vom Shopadministrator frei benannt werden. Für diese kann er wählen, ob sie zum Kunden (z.B Kundennummer, Geburtsdatum,..) oder zur Bestellung (z.B. gewünschter Liefertermin,..) gespeichert werden.

Alle Attribute können im Shop bei Bedarf darauf getestet werden, ob der Shopkunde das Feld auch ordnungsgemäss ausgefüllt hat. Diese Überprüfung besteht darin, zu testen, ob im entsprechendem Feld etwas ausgefüllt wurde (kein Leerstring). Die Überprüfung der E-Mail-Adresse ist wesentlich umfangreicher und schliesst Fehleingaben nahezu aus.

Da für jedes Attribut gewählt werden kann, ob es verwendet und überprüft werden soll, sind eine riesige Anzahl Kombinationen möglich. Wir mussten also die Ausgabe der Felder, sowie die Überprüfung der gemachten Eingaben dynamisch gestalten.

Zur Überprüfung auf korrekte Eingabe haben wir eine JavaScript Funktion erstellt, die alle Überprüfungsmöglichkeiten beinhaltet:

- keine Überprüfung der Eingabe
- Überprüfung, ob ein Wert eingegeben wurde (kein Leerstring übergeben)
- Überprüfung, ob es sich um eine E-Mail Adresse handelt

Durch den glücklichen Umstand, dass man Formularfeldinhalte eines HTML-Formulars mit JavaScript wie ein Array ansprechen kann, wird ein Zähler laufen gelassen, mit dessen Hilfe jedes Formularfeld auf seine Korrektheit überprüft wird. Damit die Funktion weiss, welches Feld wie zu behandeln ist, wird mit der Ausgabe der HTML-Seite ein JavaScript-Array per PHP erzeugt, der für jedes Feld beinhaltet, wie es überprüft werden soll. Dieser String wird der JavaScript Funktion als Argument übergeben.

Bei der Ausgabe der Bestellung an den Browser, werden die Adressdaten in einer zweispaltigen Tabelle dargestellt. Damit keine Löcher, durch nicht verwendete Adressattribute entstehen, wird auch diese Tabelle dynamisch erzeugt. Ist schlussendlich nur noch ein Adresselement auf der letzen Zeile, wird diese durch eine Dummy-Zelle ergänzt.

 © 2001 R. Glanzmann & J. Fontanil

2.11 CRM – Customer Relationship Management

Customer Relationship Management ist ein beliebig dehnbarer Begriff und momentan ein Modewort. Wir haben in unserem Shop ein einfaches Customer Relationship Management.

Von der Feedback Analyse (Kapitel 1.4) her, teilten wir die Customer Relationship Management Vorschläge und unsere eigenen Ideen in zwei Kategorien ein:

CRM 1: Eigenes Login
Automatisch ausgefüllte Adresse
Login zwingend / nicht zwingend - -> Händlermodus

CRM 2: Persönlichen Ansprechpartner konfigurieren können
Rechnung
Lieferschein drucken können
Bestellungsfortschritt abrufbar (versendet, ...)

Das CRM2 mussten wir nach einer ersten Aufwandschätzung aufgrund seiner Komplexität und den benötigten weiteren Funktionen weglassen. Wir schauten uns also das CRM 1-Paket genauer an. Nach ersten Designentscheiden haben wir herausgefunden, dass wir den Händlermodus nur implementieren könnten, wenn wir eine PHP-Authentifizierung an die Session binden würden. PHP-Auth der PHPLib wäre ein Weg um dies zu implementieren. Wir haben also auch hier gesehen, dass der Aufwand nicht zu unterschätzen ist und haben uns daraufhin dafür entschieden, vorerst nur die Grundelemente des CRM 1-Pakets zu implementieren.

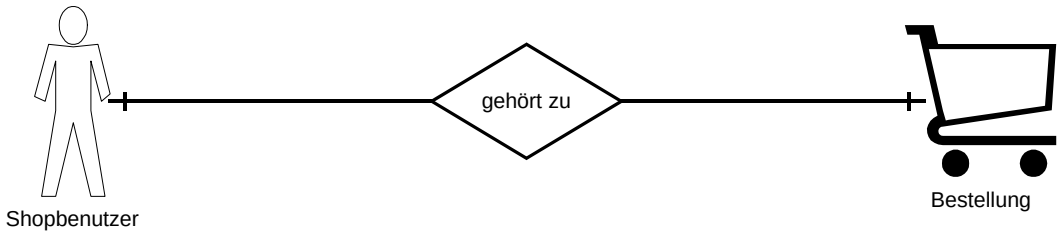
2.11.1 Welche Funktionen bietet das CRM

Wir wollten dem Kunden die Möglichkeit geben, seinen Kundendatensatz persistent zu speichern. Diese Grundfunktionalität erforderte viele weitere Funktionen und Anpassungen.

- Kunde kann sich einloggen oder auch nicht (freie Wahl).
- Kunde kann sein Passwort vergessen und es sich per E-Mail senden lassen.
- Kunde wird nach einmaligem Login wieder erkannt, bis er seinen Browser schliesst.
- Kunde kann seine Daten verändern (Datenbank-Update).

Ein detailliertes Ablaufdiagramm des CRM's findet man im Kapitel 1.7.3 (Tests).

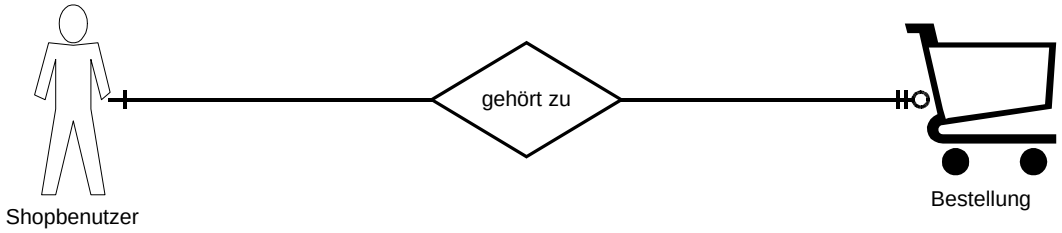
Ein weiteres Problem war, dass es durch das Einführen des CRM's persistente und temporäre Kunden gibt. Wenn ein Kunde sich nicht beim Shop anmeldet, muss sein Kundendatensatz nach der Bestellung wieder gelöscht werden, aber nur dann, wenn das Bestellungsmanagement ausgeschaltet ist. Wenn man das Bestellungsmanagement einschaltet, gibt es bei einem temporären Kunden eine Beziehung zu seiner Bestellung: Ein temporärer Kunde ist an eine Bestellung gebunden und kann nur maximal eine Bestellung haben:



Bei persistenten Kunden hingegen ist die Bestellung an einen Kunden gebunden. Wenn man

 © 2001 R. Glanzmann & J. Fontanil

hier eine Bestellung löscht, soll natürlich nicht auch der Kunde mit gelöscht werden. Ausserdem kann ein persistenter Kunde mehrere Bestellungen haben, nicht nur eine. Zum Schluss muss noch beachtet werden, dass bei Bestellungsabschluss und ausgeschaltetem Bestellungsmanagement nur die Bestellung, nicht aber der Kunde selbst, gelöscht werden darf.



Man sieht also, dass schon die grundlegenden CRM-Funktionen einiges an Analyseaufwand benötigten, um richtig implementiert zu werden.

Der PhPepperShop bietet dem Kunden jetzt an, seine Adressdaten automatisch ausfüllen zu lassen und das sogar noch SSL geschützt, wenn der Shopbetreiber SSL einschaltet.

 © 2001 R. Glanzmann & J. Fontanil

3. Manuals

3.1 Installation / Deinstallation

3.1.1 Automatisierte Installation

Nach unseren Erfahrungen mit dem „Webshop Framework in PHP“, mussten wir eingestehen, dass die Installation für viele Benutzer eine grosse Hürde darstellte. Weiter musste die Auflage erfüllt werden, dass mehrere PhPepperShops auf ein und demselben MySQL-DBMS gleichzeitig laufen können. Wir haben uns deshalb für eine automatisierte Installation des PhPepperShops entschieden.

3.1.2 Ablauf

Wenn man einen PhPepperShop installieren will, muss man wie folgt vorgehen:

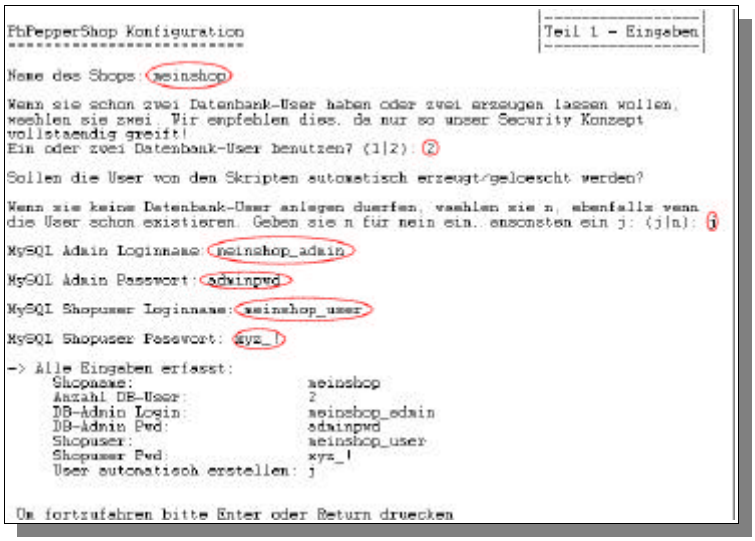
1. Entpacken des Archives in einen Ordner innerhalb des Webverzeichnisses (z.B. htdocs):
   ```
   tar xvfz PhPepperShop_xyz.tar.gz
   ```
2. Ins erzeugte Verzeichnis wechseln und das Installations-Script config.pl starten:
   ```
   perl ./config.pl
   ```

   Nun werden im Wesentlichen *fünf Schritte* abgearbeitet:

   - Eingabedaten erfassen (*siehe Punkt 3*)
   - Angepasste Shop-Verzeichnisstruktur und Dateien erzeugen
   - Angepasste Konfigurationsdateien erzeugen und ablegen:
     - `shopname_del.sql` Shop Datenbank und allfällige User entfernen
     - `shopname_create.sql` Shop Datenbank und User erstellen
     - `shopname_insert.sql` Demo-Shop und Shop-Settings einfügen
     - `initialize.php` User-Datenbankverbindung aufnehmen
     - `ADMIN_initialize.php` Administrator Datenbankverbindung aufnehmen
   - Benötigte Softlinks und Kopie des CSS-Files erstellen (Netscape Kompatibilität)
   - Angepasste Shop-Datenbank und User im MySQL DBMS erstellen

3. Erläuterung der nötigen Eingaben:

   Der „Name des Shops“ wird dazu verwendet, dem Shop einen eindeutigen Namen zu geben. Auch die Datenbank und das Shop-Verzeichnis werden so heissen.

   
   PhPepperShop Konfiguration
Teil 1 - Eingaben
Name des Shops: meinshop
Wenn sie schon zwei Datenbank-User haben oder zwei erzeugen lassen wollen, waehlen sie zwei. Wir empfehlen dies, da nur so unser Security Konzept vollstaendig greift!
Ein oder zwei Datenbank-User benutzen? (1|2): 2
Sollen die User von den Skripten automatisch erzeugt/geloescht werden?
Wenn sie keine Datenbank-User anlegen duerfen, waehlen sie n, ebenfalls wenn die User schon existieren. Geben sie n für nein ein, ansonsten ein j: (j|n): j
MySQL Admin Loginname: meinshop_admin
MySQL Admin Passwort: adminpwd
MySQL Shopuser Loginname: meinshop_user
MySQL Shopuser Passwort: xyz_!
-> Alle Eingaben erfasst:
Shopname: meinshop
Anzahl DB-User: 2
DB-Admin Login: meinshop_admin
DB-Admin Pwd: adminpwd
Shopuser: meinshop_user
Shopuser Pwd: xyz_!
User automatisch erstellen: j
Um fortzufahren bitte Enter oder Return druecken

   Die Frage nach „ein oder zwei Datenbank Usern“ spricht das Problem vieler Benutzer an, welche von ihrem Provider nur einen User zugewiesen bekommen und auch auf Anfrage keinen weiteren User erhalten. Wenn man hier '1' eintippt, muss man sich bewusst sein, dass das Sicherheitskonzept des PhPepperShops etwas untergraben wird. Das heisst im Klartext, vom Datenbank Management System her gesehen, könnte

 © 2001 R. Glanzmann & J. Fontanil

nun auch jeder Shopbenutzer Artikel oder Kategorien löschen.

Mit der Frage, ob die angegebenen User automatisch erzeugt, respektive wieder gelöscht werden sollen, wird dem Fakt nachgekommen, dass viele Benutzer von ihrem Provider einen vordefinierten MySQL-User zugesprochen bekommen und diesen nicht weiter verändern können. PhPepperShop richtet dann (optional) einfach seine Datenbank ein, verzichtet aber darauf, den / die Datenbank-User in MySQL anzumelden, resp. bei einer Deinstallation diese zu entfernen.

Hat man sich für zwei Datenbank Benutzer entschieden, kann man diese nun angeben. Mit dem 'Admin Loginname' ist der Datenbank-User gemeint, mit welchem der Shop jeweils für Administrator Aufgaben eine Verbindung zu MySQL aufbaut. Der 'Shopbenutzer' ist der Datenbank-User, welcher von den Kunden des Shops benutzt wird, um mit der Datenbank zu kommunizieren.

Muss man sich auf einen User beschränken, so werden der Administrator- und der User-Account gleichgesetzt.

Die Namen und Passwörter werden von regulären Ausdrücken überprüft. Sie müssen mindestens vier Zeichen lang sein (maximal 32) und es dürfen keine White-Spaces (Leerschlag, Tabulator) vorkommen.

Das sind auch schon alle nötigen Eingaben. Jetzt zeigt einem das Installations-Script lediglich noch die Status-Meldungen an und erklärt laufend, was es gerade macht.

Man wird gegen Ende der Installation noch gefragt, ob die kreierte Datenbank nun auch ins MySQL Datenbank Management System eingefügt werden soll. Auf diese Weise kann man den Shop neu installieren, die schon bestehende Shop-Datenbank aber weiterhin benutzen. Dies ist vor allem für Updates des PhPepperShops wichtig. Diese Methode funktioniert aber nur, solange bei einem künftigen Update von PhPepperShop kein Datenbankupdate vorgenommen wird.

3.1.3 Installationsmechanismus

Hier gehen wir etwas detaillierter auf die Implementation der automatischen PhPepperShop Installation ein.

Die automatisierte Installation besteht aus folgenden drei Dateien:

```
templateshop/config.pl
templateshop/config_part4.sh
templateshop/template/template_remove.pl
```

Die Datei `config.pl` ist, wie die Datei-Endung schon impliziert, ein Perl-Script. Dieses Script bestreitet den Hauptteil der Installation. Das Shell-Script `config_part4.sh` kümmert sich v.a. um die Erstellung von benötigten Softlinks. Zur Deinstallation des jeweiligen PhPepperShops benötigt man ein an den jeweiligen Shop angepasstes Perl-Script, welches aus dem `template_remove.pl` erstellt wird. Dieses Script wird nach der 'Personalisierung'' im erstellten Shop unter remove.pl abgelegt.

Damit mehrere Shops auf einem DBMS installiert werden können, wird jeder PhPepperShop aus einem 'Template'-Shop erzeugt und in einem Verzeichnis gleich dem Shop Namen abgespeichert. Natürlich erhält jeder Shop seine eigene Datenbank. Auf diese Weise kann man mehrere Shops übersichtlich verwalten.

Wir haben bei der Implementierung des Installationstools auch gleich noch folgende Probleme gelöst: Ein-User-Betrieb, schon bestehende(-r) User und schon bestehende Datenbank. Wäh-

 © 2001 R. Glanzmann & J. Fontanil

rend der Installation kann man entscheiden, welche Konfiguration man für seinen Shop verwenden möchte.

3.1.4 Erste Konfigurationen

Zuerst muss man das Admin Verzeichnis (<shopdir>/shop/Admin/) per htaccess schützen. Sonst hat jeder, der den Link kennt, Zugriff auf den Administrationsteil des Shops. Wie man hier *genau* vorgeht, ist von Installation zu Installation verschieden und kann nicht allgemein beantwortet werden. Ausserdem ist es Sache des Providers. Wir haben unsere Erkenntnisse in Form einer kleinen Anleitung auf der PhPepperShop-Homepage (www.phpeppershop.com) zum download freigegeben. Wir möchten hier aber nochmals darauf hinweisen, dass unsere Anleitung nicht allgemein gültig ist.

Nun kann man auf den Stern links neben dem Button "Warenkorb anzeigen" klicken. Hier kommt man in den Administrationsmodus. Der Stern lässt sich im Layout Management natürlich abschalten, aber zu Beginn, wenn man noch öfters in den Administrationsmodus geht, ist er durchaus praktisch. Zuerst sollten alle Angaben in den allgemeinen Shopeinstellungen überprüft und gegebenenfalls angepasst werden. Insbesondere sollte man hier die Adressinformationen und die Allgemeinen Geschäftsbedingungen korrekt ausfüllen. Danach kann man die Versandkostensettings seinen Bedürfnissen anpassen. Alle weiteren Einstellungen können im Kapitel 3.3.2, „Bedienung des Administrationstools“ nachgelesen werden.

3.1.5 Deinstallation des Shops

Um einen PhPepperShop wieder sauber zu deinstallieren, müssen alle Shop-Dateien im Dateisystem entfernt und die Datenbank gelöscht werden. Auch zur Deinstallation haben wir ein automatisiertes Tool in Perl geschrieben.

Um die Deinstallation zu starten, muss man im Shopverzeichnis folgenden Systemaufruf starten:

```
perl ./remove.pl
```

Nachdem man bestätigt hat, dass man den Shop auch wirklich deinstallieren will, kann man angeben, ob auch die Datenbank (User und Shop-Daten in der Datenbank) deinstalliert werden soll. Bei einem Update des Shops kann man so den Shop deinstallieren und die Datenbank beibehalten. Man muss hierbei aber berücksichtigen, dass die Bilder des Shops (Hintergrundbilder, Buttons, etc.) nicht in der Datenbank gespeichert sind und gelöscht werden.

Der Shop wird komplett deinstalliert. Alle vom Shop verwendeten Verzeichnisse und Dateien sind danach gelöscht.

 © 2001 R. Glanzmann & J. Fontanil

3.2 Backup und Restore mit MySQL

Hier zeichnet sich nun der Vorteil unseres Konzepts aus: Der *ganze* Shop ist in der Datenbank abgebildet, inkl. Konfiguration, Daten und Bilder. Wenn man also ein Backup der Datenbank erstellt, hat man zugleich auch ein Backup des Shops, mit Ausnahme der Buttons und Hintergrundbilder des Shops. Diese sind aus Performancegründen nicht in der Datenbank, sondern im Dateisystem untergebracht. Die Daten liegen aber komplett in der Datenbank.

Wir gehen hier spezifisch auf MySQL als Datenbank ein, weil sie als reine Webdatenbank am meisten benutzt wird. Man kann ein Datenbank Backup auch über kombinierte SELECT-Statements, kompatibel mit allen Datenbanken, über alle Tabellen machen. Wir haben uns dies schon überlegt, aber aus Zeitgründen nicht mehr implementieren können.

Das komplette Backup eines PhPepperShops umfasst genau genommen zwei Teile:

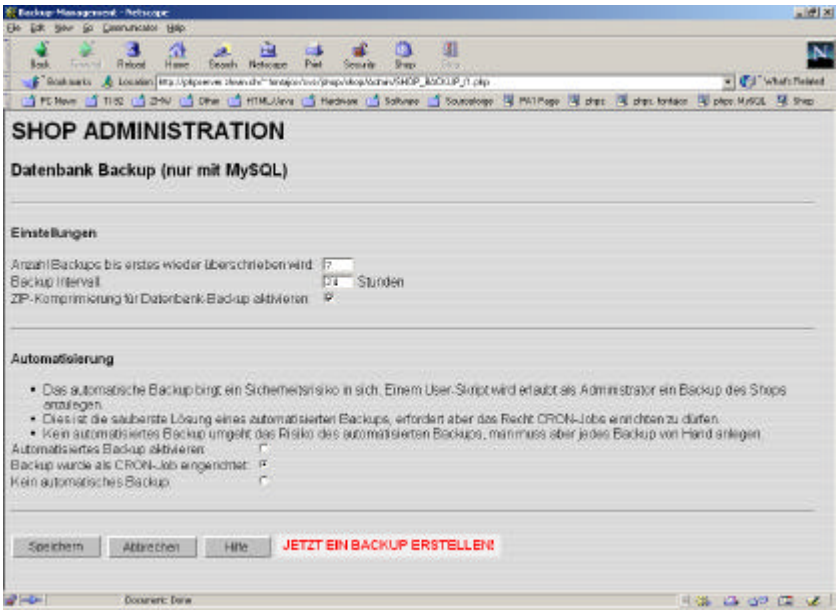
1. Backup von User-Buttons und Hintergrundbildern
2. Backup der Daten des Shops (Datenbank)

3.2.1 Automatisiertes Datenbank-Backup

Das automatisierte Datenbank-Backup erlaubt es ein Backup-Set anzulegen, welches optional automatisch aktualisiert wird. Wir möchten hier ausdrücklich darauf hinweisen, dass es sich hierbei um ein Datenbank-Backup handelt. **Die User-Buttons und Hintergrundbilder werden NICHT mit diesem Backup nicht gespeichert**! Diese müssen immer noch von Hand gespeichert werden (siehe weiter unten).

Der Shop bietet im Administrationsmodus unter dem Punkt Datenbank-Backup ein Untermenü an. Von hier aus lässt sich das automatisierte Datenbank Backup konfiguieren, ein Datenbank-Backup erstellen oder ein Restore starten.

Wenn man im Untermenü auf Backup klickt, kann man das automatisierte Backup konfigurieren. Hier kann man auch direkt ein Backup machen (Button 'JETZT EIN BACKUP ERSTELLEN').



In den Einstellungen kann man den Umfang des Backup-Sets und das Backup-Intervall festlegen. Optional lässt sich noch die ZIP-Komprimierung aktivieren. Die ZIP-Komprimierung

 © 2001 R. Glanzmann & J. Fontanil

funktioniert allerdings nur wenn die ZLib-Library eingebunden ist. Da ein Shop 24h offen ist und 7 Tage in der Woche erreichbar ist, empfehlen wir ein Backup-Set von sieben Dateien anzulegen und das Backup-Intervall auf 24 Stunden einzustellen. Wenn es möglich ist die ZIP-Komprimierung zu aktivieren, sollte dies auch gemacht werden.

Wenn man von PHP und Automatisierung redet, gibt es grundsätzlich ein Problem: Man hat das Problem, dass man ein programmiertes Ereignis nicht einfach nach Ablauf einer gewissen Frist aufrufen kann. Wir bieten deshalb unter dem Menüpunkt 'Automatisierung' drei verschiedene Einstellungsmöglichkeiten:

1. *Automatisiertes Backup aktivieren:* Hierbei wird der Datei `index.php` ein Backup Aufruf eingefügt (include von `<shopdir>/shop/Admin/ADMIN_backup.php`). Diese Datei überprüft ob es Zeit ist ein weiteres Backup anzulegen und tut dies falls notwendig auch gleich.

   Bei diesem Ansatz gibt es eine Sicherheitslücke. Es ist der einzige Ort im Shop bei welchem wir mit unserer Sicherheits-Policy brechen mussten. Ein USER-Script ruft ein ADMIN-Script auf. Schliesslich ist das Anlegen eines Datenbank-Backups Administrator-Sache. Dies ist also ein Kompromiss zwischen Usability und Security. Wir empfehlen hier dringend, dass man wenn immer möglich einen CRON-Job einrichten sollte, welcher dieses Backup periodisch aufruft (Aufruf: `php ADMIN_backup.php`).

   Der Aufruf des Backups in der index.php Datei ist 'atomar'. Es handelt sich um einen einzigen Aufruf welcher entweder komplett ausgeführt wird oder nicht aufgerufen wird.

2. *Backup wurde als CRON-Job eingerichtet:* Dies ist wohl die eleganteste Methode des automatisierten Backups und zugleich auch die sicherste. Man muss jetzt per Telnet oder besser SSH einen CRON-Job einrichten. Dieser muss lediglich im Backup-Intervall die Datei `ADMIN_backup.php` aufrufen (im Admin Unterverzeichnis). Das Backup-Set wird automatisch gepflegt.

3. *Kein automatisches Backup:* Es wird kein automatisiertes Backup vorgenommen. Jedes Backup muss von Hand angelegt werden. Wer keine Kompromisse betreffend der Security eingehen will aber auch keine CRON-Jobs einrichten darf (manche Provider verbieten dies), dem raten wir zu dieser Einstellung.

Dateinamen

Der Datenbank-Backup legt das aktuelle Backup in der Datei `0.sql[.gz]` ab. Gibt es ein Backup-Set grösser als eins, so wird der Dateiname der schon existierenden Backups um eins dekrementiert. Das neueste Backup ist also immer 0. Das älteste Backup bei einem Backup-Set von 5 wäre also `4.sql[.gz]`.

Es wird immer zuerst umbenannt und danach das neue Backup `0.sql[.gz]` erzeugt.

phpMyBackup

Die Grundlage unserer automatisierten Backuplösung stammt aus dem Programm phpMyBackup von Holger Mauermann (mauermann@nm-service.de). Er hat dieses vielversprechende PHP-Script für den Backup und Restore einer MySQL Datenbank geschrieben. Wir haben als Grundlage phpMyBackup in der Version 0.4 Beta benutzt. Das Script kann unter http://www.nm-service.de/phpmybackup gefunden werden. Wir möchten Holger an dieser Stelle nochmals recht herzlich für die Erlaubnis, sein Programm benutzen zu dürfen, danken.

 © 2001 R. Glanzmann & J. Fontanil

3.2.2 Restore eines Datenbank-Backups

Wenn man im Backup-Menü auf 'Restore' klickt kann ein Datenbank-Backup angesehen oder zurück gelesen werden. Mit 'view/download' kann man die SQL-Datei ansehen. Achtung: Wenn viele Bilder in der Datenbank sind, so werden diese mit übertragen!

3.2.3 Backup der Shopdateien und Shopbilder von Hand

Hinweis: Wörter in dreieckigen Klammern stellen Bezeichner dar. Man muss bei folgenden Befehlen die Bezeichner durch entsprechende Angaben ersetzen. Beispiel:

```
<shopname>/Bilder --> pizzashop/Bilder
```

Als Teil des Layout Managements kann man alle für den Shopbenutzer sichtbaren Buttons und die Hintergrundbilder der drei Frames im Administrationstool ändern. Diese Grafiken werden in eigens dafür vorgesehenen Dateien in den Verzeichnissen `<shopdir>/shop/Buttons/` und `<shopdir>/shop/Bilder/` gespeichert.

Um die für das Backup notwendigen Befehle eingeben zu können, öffnet man eine Telnet oder besser eine SSH Konsole zum betreffenden Webserver.

Wir schlagen vor, ein Backupverzeichnis ausserhalb des Webverzeichnisses anzulegen und dort die Backupdaten zu hinterlegen. Wenn man regelmässig Backups anlegen will, was wir als sehr sinnvoll erachten, empfiehlt es sich, hier eine bezeichnende Verzeichnis-Namensgebung zu benutzen. Darin erstellt man folgende Verzeichnisse:

```
mkdir shopbackup
mkdir shopbackup/Bilder
mkdir shopbackup/Buttons
mkdir shopbackup/Datenbank
```

Dann kopiert man die Shopbuttons, Shopbilder und Datenbank SQLs. Dazu geht man zuerst ins Shopverzeichnis und gibt folgende Befehle ein:

```
cp ./shop/Bilder/* <Pfad und Backupverzeichnis>/Bilder/
cp ./shop/Buttons/* <Pfad und Backupverzeichnis>/Buttons/
cp ./shop/database/* <Pfad und Backupverzeichnis>/Datenbank/
```

3.2.4 Backup der Shop Datenbank von Hand

Ein Backup einer MySQL Datenbank lässt sich am einfachsten über das mitgelieferte MySQL-Tool `mysqldump` erledigen. Es befindet sich im Unterverzeichnis `bin/` der MySQL-Datenbank.

Das Backup der Shopdatenbank wird wie folgt ausgeführt:

```
mysqldump -c -u<Username> -p <shopname> > <Pfad und Backupverzeich-
nis>/Datenbank/<shopname>backup<datum>.sql
```

Beispiel:

```
mysqldump -c -uPizzauser -p pizzashop > ~/pizzashopbackup_27102001/Da-
tenbank/pizzashopbackup27102001.sql
```

Nun ist ein Backup der gesamten Shopdatenbank, inklusive allen persistenten Kunden, Bestellungen (falls das Bestellungsmanagement eingeschaltet ist) und Artikeln mit ihren Bil-

 © 2001 R. Glanzmann & J. Fontanil

dern erstellt worden.

In verschiedenen Publikationen wurde darauf hingewiesen, dass gewisse Provider ihren MySQL-Benutzern aus Sicherheitsgründen das Verwenden dieses Programms verbieten. Wir haben uns überlegt, dass man ein Backup einer beliebigen Datenbank auch aus kumulierten SELECT-Statements zusammenstellen kann. Leider blieb uns zu wenig Zeit, um diesen Gedanken in Form eines Backuptools noch rechtzeitig in diese Diplomarbeit einfliessen zu lassen. Wir haben uns aber entschieden, baldmöglichst ein Backup-Utility nachzureichen.

3.2.5 Restore von Hand

Auch hier: Wörter in dreieckigen Klammern stellen Bezeichner dar. Man muss bei folgenden Befehlen die Bezeichner durch entsprechende Angaben ersetzen. Beispiel:

```
<shopname> --> pizzashop
```

Der Restore lässt sich in vier Teilschritte zerlegen.

1. Man muss einen frisch installierten / upgedateten Shop haben, welcher betriebsbereit ist.
2. Restore der Bilder und Buttons:

   Im Backupverzeichnis:

   ```
   cp ./Bilder/* <Pfad zu Shopverzeichnis>/shop/Bilder/
   cp ./Buttons/* <Pfad zu Shopverzeichnis>/shop/Buttons/
   mysql -u<Username> -p < ./Datenbank/<shopname>_del.sql
   mysql -u<Username> -p < ./Datenbank/<shopname>_create.sql
   ```

   Jetzt kopiert man die Shop-Buttons und Hintergrundbilder in den neuen Shop zurück und löscht die Default Shop Datenbank. Danach kreiert man eine neue, leere Datenbank.
3. Restore der Daten aus dem Backup der Shop Datenbank:

   Nun muss man die Shop Datenbank zurück lesen:

   ```
   mysql -u<Username> -p <shopname> < <shopname>backup<datum>.sql
   ```
4. Persönliche CSS-Einstellungen aktivieren: Jetzt ist die Shopdatenbank zwar wieder zurück gelesen, die CSS-Einstellungen werden aber erst nach einem Speichern der Einstellungen im Layout Management neu übernommen (shopstyles.css wird neu generiert). Deshalb muss man zuerst im Administrationstool ins Layout Management gehen und auf Speichern klicken.

 © 2001 R. Glanzmann & J. Fontanil

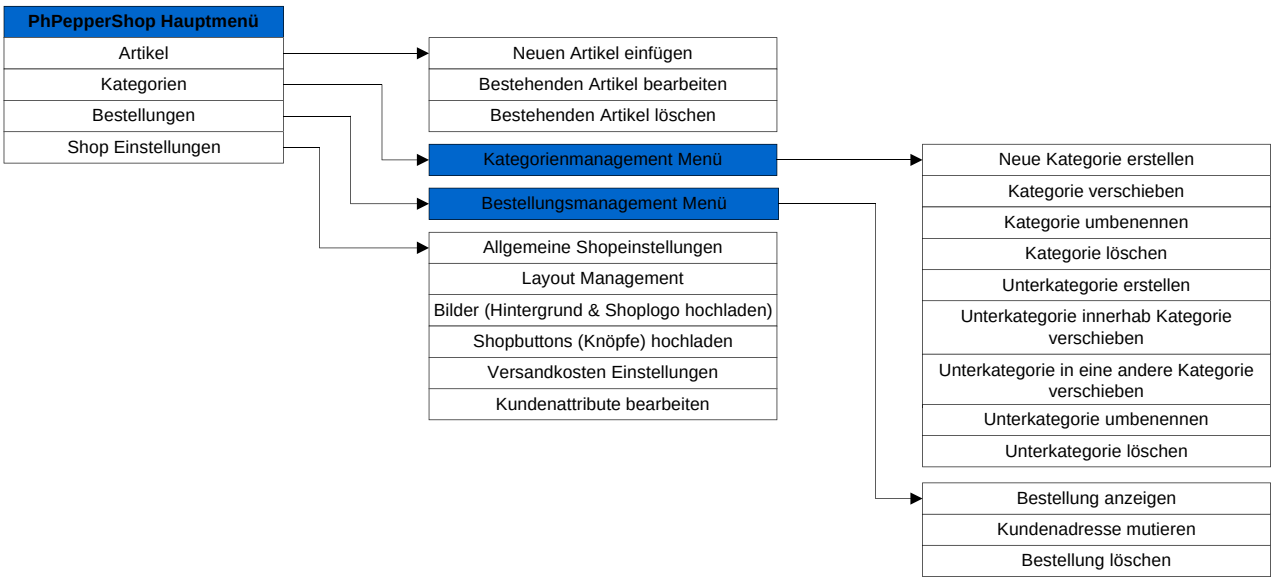
3.3 Bedienung

3.3.1 Bedienung des Shops

Maximale Usability war eines unserer Hauptziele. Da jede Person, die über einen Zugriff zum Internet verfügt, zum Benutzer unserer Software werden kann, musste der Shop selbst erklärend aufgebaut werden. Wir hätten unser Ziel weit verfehlt, wenn wir hier eine Bedienungsanleitung abdrucken müssten. Denn wer hat schon Lust, eine Bedienungsanleitung zu lesen, um einen Einkauf tätigen zu können.

3.3.2 Bedienung des Administrationstools

Folgende Grafik zeigt die Menüstruktur mit den möglichen Aktionen auf:



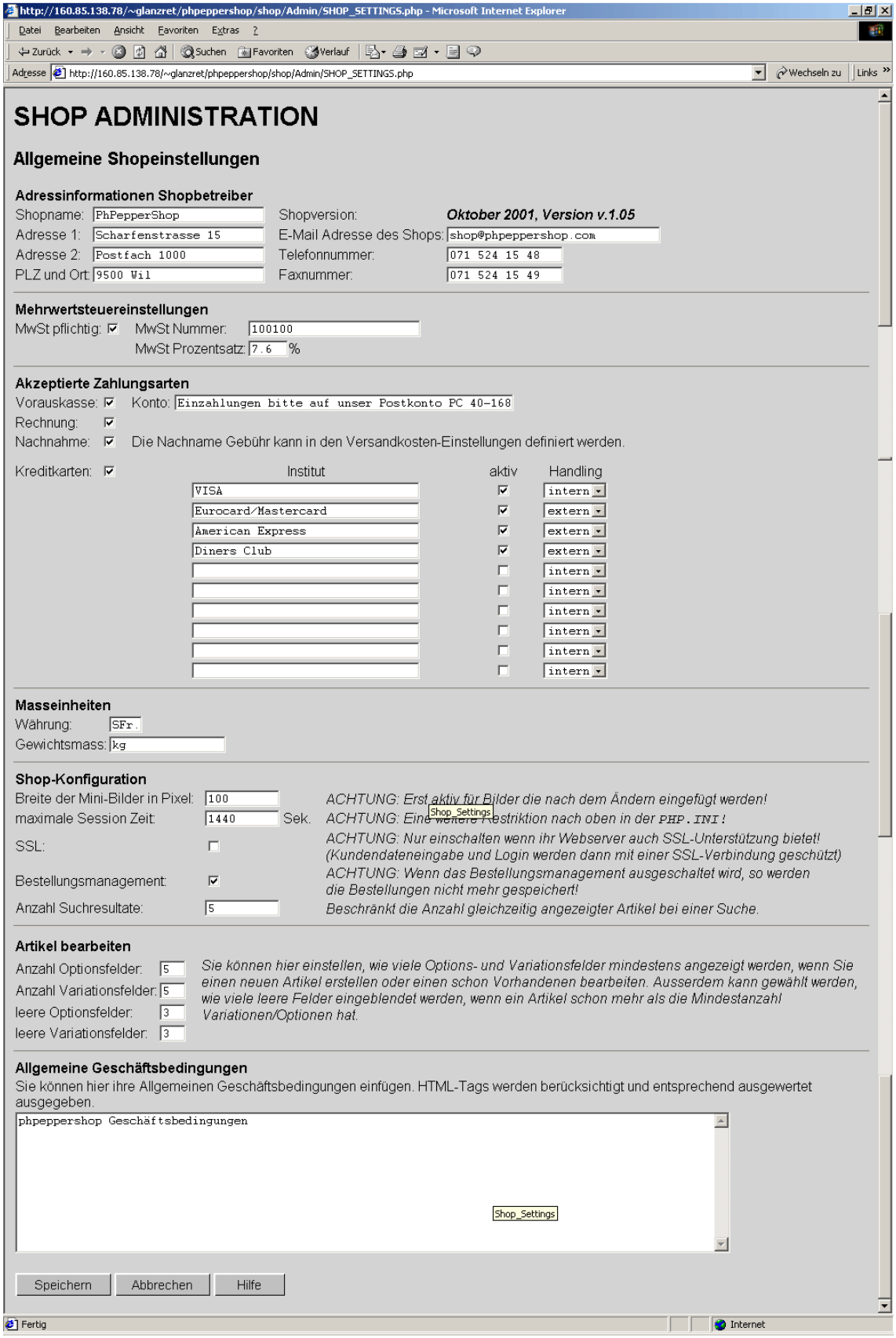
Das Artikelmanagement (Artikel erstellen, bearbeiten und löschen), das Kategorienmanagement sowie das Bestellungsmanagement sind selbst erklärend aufgebaut und erlauben es auch nicht so versierten Benutzern, den Shop zu administrieren.

Ein kleines Manko des Kategorienmanagements soll hier noch erwähnt werden: Fügen Sie eine Unterkategorie in eine Kategorie ein, die keine weiteren Unterkategorien enthält, werden Sie gefragt, wo die neue Unterkategorie eingefügt werden soll. Da sich keine weiteren Unterkategorien in dieser Kategorie befinden, ist diese Frage eigentlich überflüssig. Es wird nur ein Knopf „hier einfügen“ angezeigt. Diesen müssen Sie drücken, um die neue Unterkategorie zu erstellen.

Allgemeine Shopeinstellung

In diesem Formular können Sie die meisten Shopeinstellungen tätigen.

 © 2001 R. Glanzmann & J. Fontanil


http://160.85.138.78/~glanzret/phpeppershop/shop/Admin/SHOP_SETTINGS.php - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Verlauf
Adresse http://160.85.138.78/~glanzret/phpeppershop/shop/Admin/SHOP_SETTINGS.php
Wechseln zu Links
SHOP ADMINISTRATION
Allgemeine Shopeinstellungen
Adressinformationen Shopbetreiber
Shopname: PhPepperShop
Adresse 1: Scharfenstrasse 15
Adresse 2: Postfach 1000
PLZ und Ort: 9500 Wil
Shopversion: Oktober 2001, Version v.1.05
E-Mail Adresse des Shops: shop@phpeppershop.com
Telefonnummer: 071 524 15 48
Faxnummer: 071 524 15 49
Mehrwertsteuereinstellungen
MwSt pflichtig:
MwSt Nummer: 100100
MwSt Prozentsatz: 7.6 %
Akzeptierte Zahlungsarten
Vorauskasse: Konto: Einzahlungen bitte auf unser Postkonto PC 40-168
Rechnung:
Nachnahme: Die Nachname Gebühr kann in den Versandkosten-Einstellungen definiert werden.
Kreditkarten:
Institut
aktiv
Handling
VISA intern
Eurocard/Mastercard extern
American Express extern
Diners Club extern
Masseinheiten
Währung: SFr.
Gewichtsmass: kg
Shop-Konfiguration
Breite der Mini-Bilder in Pixel: 100
ACHTUNG: Erst aktiv für Bilder die nach dem Ändern eingefügt werden!
maximale Session Zeit: 1440 Sek.
Shop_Settings
ACHTUNG: Eine weitere Restriktion nach oben in der PHP.INI!
SSL:
ACHTUNG: Nur einschalten wenn ihr Webserver auch SSL-Unterstützung bietet! (Kundendateneingabe und Login werden dann mit einer SSL-Verbindung geschützt)
Bestellungsmanagement:
ACHTUNG: Wenn das Bestellungsmanagement ausgeschaltet wird, so werden die Bestellungen nicht mehr gespeichert!
Anzahl Suchresultate: 5
Beschränkt die Anzahl gleichzeitig angezeigter Artikel bei einer Suche.
Artikel bearbeiten
Anzahl Optionsfelder: 5
Anzahl Variationsfelder: 5
leere Optionsfelder: 3
leere Variationsfelder: 3
Sie können hier einstellen, wie viele Options- und Variationsfelder mindestens angezeigt werden, wenn Sie einen neuen Artikel erstellen oder einen schon Vorhandenen bearbeiten. Ausserdem kann gewählt werden, wie viele leere Felder eingeblendet werden, wenn ein Artikel schon mehr als die Mindestanzahl Variationen/Optionen hat.
Allgemeine Geschäftsbedingungen
Sie können hier ihre Allgemeinen Geschäftsbedingungen einfügen. HTML-Tags werden berücksichtigt und entsprechend ausgewertet ausgegeben.
phpeppershop Geschäftsbedingungen
Shop_Settings
Speichern
Abbrechen
Hilfe
Fertig
Internet

 © 2001 R. Glanzmann & J. Fontanil

Adressinformationen Shopbetreiber

Füllen Sie in diesem Abschnitt Ihre Adresse ein.

Shopname:	Geben Sie in dieses Feld Ihren Shopnamen ein. Dieser wird an verschiedenen Stellen des Shops verwendet und erscheint auch auf der E-Mail-Bestellbestätigung für den Kunden.
E-Mail Adresse:	Die E-Mail-Adresse müssen Sie unbedingt angeben! An diese Adresse werden Ihnen die Bestellungen Ihrer Kunden gemailt.

Mehrwertsteuereinstellungen

Falls Sie mehrwertsteuerpflichtig sind, können sie hier Ihre Mehrwertsteuerdaten eingeben.

MwSt pflichtig:	Hier können Sie die MwSt. Funktion ein-/ausschalten. Ist die Mehrwertsteuerfunktion eingeschaltet, wird auf der Bestellung eingeblendet, dass die Preise inklusive Mehrwertsteuer sind. Ist diese Funktion ausgeschaltet, haben die folgenden zwei Felder keine Bedeutung.
MwSt Nummer:	Geben Sie hier Ihre MwSt Nummer. Diese wird dem Kunden in seiner Bestellbestätigung mitgegeben.
MwSt Prozentsatz:	Diese Angabe wird an den gleichen Orten ausgegeben, wie die MwSt Nummer. Sie hat auf die Berechnung der Preise keinen Einfluss!

Akzeptierte Zahlungsarten

In diesem Abschnitt können Sie wählen, welche Zahlungsarten Sie in Ihrem Shop akzeptieren wollen.

Vorauskasse:	Aktivieren Sie diese Checkbox, wenn Sie die Bezahlung per Vorauskasse aktivieren wollen.
Konto:	Dieser Text wird dem Kunden in der Bestellbestätigung ausgegeben, falls dieser die Zahlungsart Vorauskasse wählt. Sie können hier zum Beispiel Ihre Kontoinformationen eingeben, damit der Kunde das Geld darauf einzahlen kann.
Rechnung:	Zahlungsart Rechnung aktiviert/nicht aktiviert
Nachnahme:	Zahlungsart Nachnahme aktiviert/nicht aktiviert. Wählt ein Kunde die Zahlungsart Nachnahme, wird ihm ein Pauschalpreis für die entstehenden Nachnahmegebühren auf den Rechnungsbetrag addiert. Die Höhe des Betrags kann im Menü Versandkosten-Einstellungen fest gesetzt werden.
Kreditkarten:	Aktivieren Sie diese Checkbox, wenn Sie die Bezahlung mit Kreditkarten aktivieren wollen.
Institut:	Geben Sie den Namen der Kreditkartenanbieter ein.
aktiv:	Mit diesen Checkboxen können Sie alle Kreditkarten getrennt voneinander aktivieren/deaktivieren.

 © 2001 R. Glanzmann & J. Fontanil

Handling:	Für jede Kreditkarte kann eingestellt werden, wie die Kreditkartendaten erfasst werden. **extern:** Die Kreditkartendaten werden von einem externen Zahlungsinstitut erfasst. **intern:** Der Kunde gibt seine Kreditkartendaten direkt im Shop ein. Wir raten Ihnen dringend, die SLL-Verschlüsselung einzuschalten!

Masseinheiten

Zur Einstellung von Länder- und Branchenspezifischen Masseinheiten

Währung:	Legt fest, welche Währung im Shop verwendet werden soll.
Gewichtsmass:	Definiert die Gewichtseinheit, die angezeigt werden soll.

Shop-Konfiguration

Verschiedene Shopeinstellungen

Breite der Mini-Bilder in Pixel:	Wenn Sie ein Artikelbild hochladen, wird davon automatisch eine kleine Voransicht erstellt, die dann im Shop angezeigt wird. Diese sogenannten Thumbnails haben alle die gleiche Breite, welche hier festgelegt werden kann.
Maximale Session Zeit:	Jedem Shopbesucher wird beim Betreten des Shops eine Identifikationsnummer zugeteilt. Sie wird nach einer gewissen Zeit wieder gelöscht. Diese Zeit können Sie hier in Anzahl Sekunden einstellen.
SSL:	Wenn Sie die SSL-Verschlüsselung aktivieren, sind alle Seiten, die ausgegeben werden, nachdem sich der Kunde eingeloggt hat, SSL-verschlüsselt. ACHTUNG!! Der Web-Server muss für SSL-Verschlüsselung eingerichtet sein, damit diese Funktion gebraucht werden kann. Wenden Sie sich bei Unklarheiten an Ihren Provider.
Bestellungsmanagement:	Ist das Bestellungsmanagement aktiviert, werden alle Bestellungen auf dem Webserver gespeichert und können vom Administrator im Menü 'Bestellungsmanagement' betrachtet oder gelöscht werden. Achtung: Wenn man das Bestellungsmanagement erst später abschaltet, werden die vorher gespeicherten Bestellungen nicht gelöscht.
Anzahl Suchresultate:	In diesem Feld können Sie eingeben, wie viele Suchresultate pro Suchresultatseite ausgegeben werden. Wenn auf eine Suchanfrage eines Kunden viele Treffer zurückgegeben werden, kann er sich einfach durch die Suchresultatseiten durchblättern.

Artikel bearbeiten

In dieser Eingabegruppe können sie bestimmen, wie viele Options-/Variaten-Eingabefelder bei der Bearbeitung/Erstellung eines Artikels angezeigt werden sollen.

 © 2001 R. Glanzmann & J. Fontanil

Anzahl Optionsfelder:	Anzahl Optionsfelder, die mindestens ausgegeben werden, wenn der Administrator einen Artikel neu erstellt oder bearbeitet.
Anzahl Variationsfelder:	Anzahl Variationsfelder, die mindestens ausgegeben werden, wenn der Administrator einen Artikel neu erstellt oder bearbeitet.
leere Optionsfelder:	Anzahl der leeren Optionsfelder, die beim Bearbeiten des Artikels ausgegeben werden, wenn ein Artikel schon viele Optionen hat.
leere Variationsfelder:	Anzahl der leeren Variationsfelder, die beim Bearbeiten des Artikels ausgegeben werden, wenn ein Artikel schon viele Varianten hat.

Allgemeine Geschäftsbedingungen

In dieses Textfeld können Sie Ihre Allgemeinen Geschäftsbedingungen einfügen. Diese werden dem Kunden zur Akzeptierung angezeigt, bevor die Bestellung abgeschickt werden kann. HTML-Tags, die sich im Text befinden, werden interpretiert. Sie können zum Beispiel einen Teil des Textes Fett formatiert ausgeben.

Versandkosten Einstellungen

In diesem Menüpunkt können Sie sehr flexibel einstellen, welche zusätzlichen Gebühren Ihren Kunden verrechnet werden.

Sie haben drei Möglichkeiten, die Versandkosten zu berechnen:

1. **nach Preis**: Die Höhe der Versandkosten wird aus der Summe des Gesamtpreises aller Artikel, die der Kunde im Warenkorb hat, berechnet. Es können vier verschiedene Preisintervalle eingegeben werden, für die jeweils ein eigenständiger Versandkostenbetrag definiert werden kann.
2. **nach Gewicht**: Die Höhe der Versandkosten wird aus der Summe des Gewichts aller Artikel, die der Kunde im Warenkorb hat, berechnet. Es können vier verschiedene Gewichtsintervalle eingegeben werden, für die jeweils ein eigenständiger Versandkostenbetrag definiert werden kann.
3. **nach Pauschale**: Der Kunde bezahlt immer den gleichen Betrag für Porto und Verpackung, egal was und wie viel er einkauft.

Bei der Versandkostenberechnung nach Gewicht und Preis können Sie für jedes Intervall bestimmen, welche Bezahlungsmöglichkeiten akzeptiert werden. Zum Beispiel können Sie die Bezahlung per Kreditkarte erst ab SFr. 50.-- erlauben, oder Bestellungen ab SFr. 300.-- nicht mehr mit einer Rechnung ausliefern.

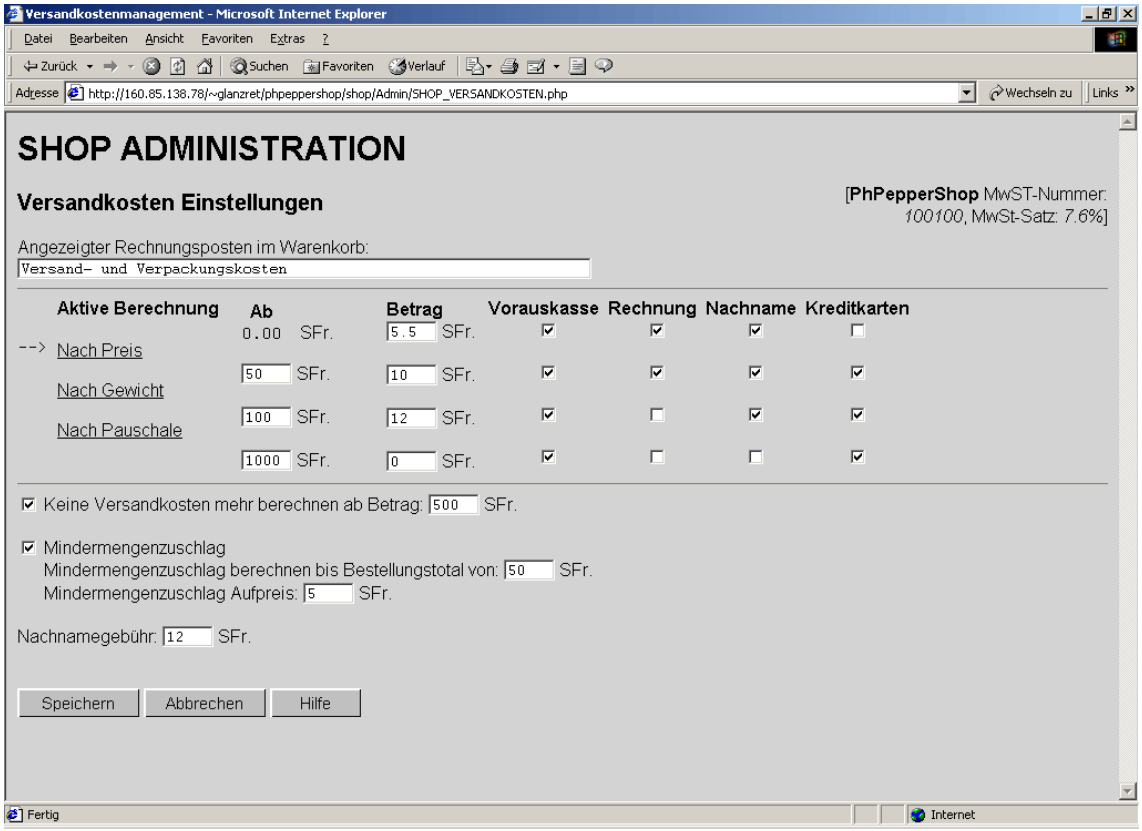
Die jeweils aktivierte Berechnungsmethode ist durch einen vorangestellten Pfeil zu erkennen.

Völlig unabhängig von den Versandkosteneinstellungen können Sie zusätzlich einen Rechnungsbtrag festlegen, bei dessen Überschreitung keine Versandkosten mehr berechnet werden.

Bei Unterschreiten eines gewissen Totalbetrags besteht die Möglichkeit, diesem einen Mindermengenzuschlag hinzuzufügen. Höhe und Betrag sind frei wählbar.

Letzer Einstellungspunkt in diesem Menü ist die Höhe einer eventuell anfallenden Nachnahmegebühr. Diese kann nur eingegeben werden, wenn Sie die Bezahlungsart „Nachnahme“ aktiviert haben.

 © 2001 R. Glanzmann & J. Fontanil


Versandkostenmanagement - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Verlauf
Adresse http://160.85.138.78/~glanzret/phpeppershop/shop/Admin/SHOP_VERSANDKOSTEN.php
Wechseln zu Links
SHOP ADMINISTRATION
Versandkosten Einstellungen
[PhPepperShop MwST-Nummer: 100100, MwSt-Satz: 7.6%]
Angezeigter Rechnungsposten im Warenkorb:
Versand- und Verpackungskosten
Aktive Berechnung Ab Betrag Vorauskasse Rechnung Nachname Kreditkarten
--> Nach Preis 0.00 SFr. 5.5 SFr.
Nach Gewicht 50 SFr. 10 SFr.
Nach Pauschale 100 SFr. 12 SFr.
1000 SFr. 0 SFr.
Keine Versandkosten mehr berechnen ab Betrag: 500 SFr.
Mindermengenzuschlag
Mindermengenzuschlag berechnen bis Bestellungstotal von: 50 SFr.
Mindermengenzuschlag Aufpreis: 5 SFr.
Nachnamegebühr: 12 SFr.
Speichern Abbrechen Hilfe
Fertig
Internet

Kundenattribute bearbeiten

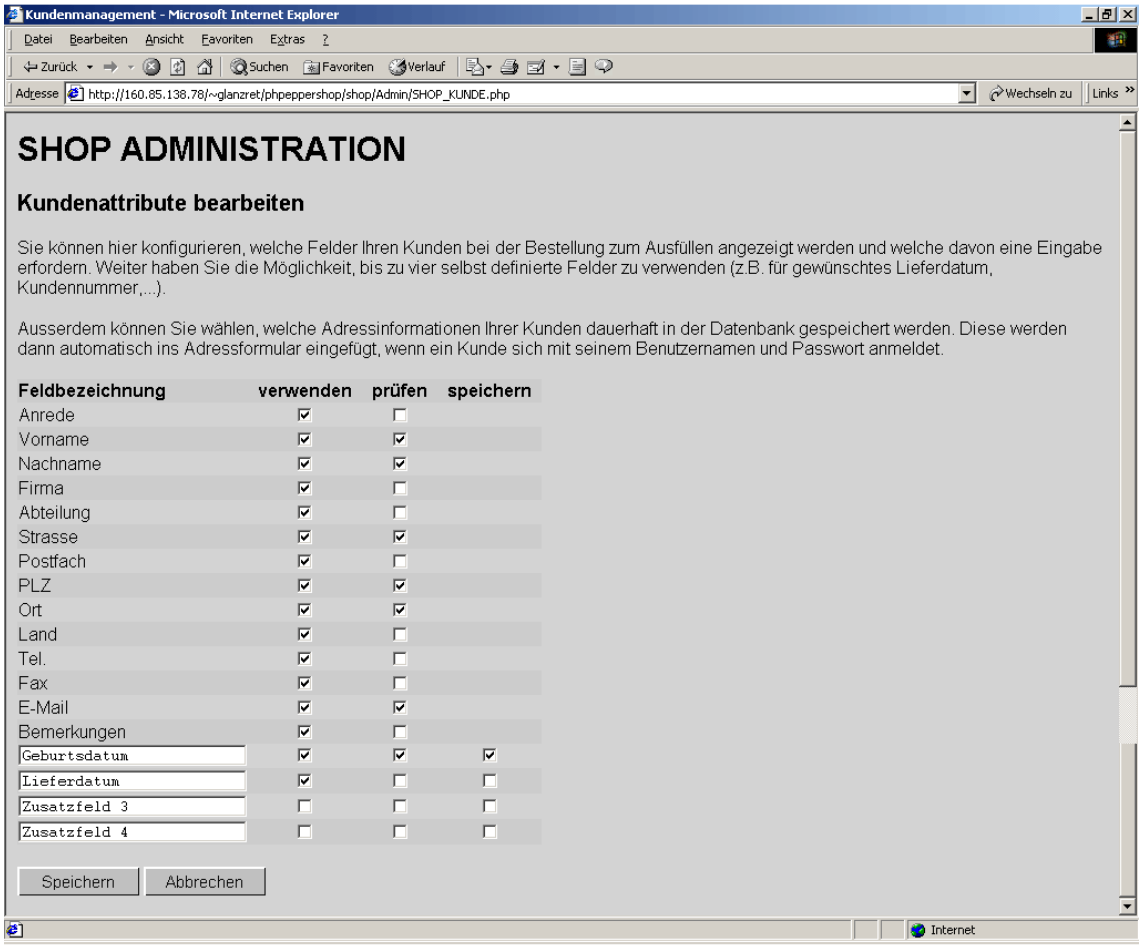
Sie können bestimmen, welche Adressinformationen Sie von Ihren Kunden erhalten möchten und welche davon unbedingt ausgefüllt werden müssen, damit die Bestellung abgesendet werden kann.

Sie haben 14 vordefinierte Felder zur Verfügung, die sie nach Belieben anzeigen (Checkbox „**verwenden**“) und überprüfen (Checkbox „**prüfen**“) lassen können.

Zusätzlich können bis zu vier Zusatzfelder aktiviert werden, deren Beschreibungstext Sie selbst bestimmen können.

Diese Felder besitzen eine zusätzliche Checkbox „**speichern**“. Ist diese aktiviert, wird das Feld in den Kundendatensatz aufgenommen. Die in diesem Feld eingegebenen Daten werden demzufolge mit den anderen Adressdaten des Kunden gespeichert. Meldet sich der Kunde beim nächsten Besuch Ihres Shops mit seinem Benutzernamen und Passwort an, wird dieses Feld automatisch ausgefüllt. Dies ist für Angaben wünschenswert, die sich nie, oder selten ändern (z.B. Kundennummer, Geburtsdatum,..). Falls Sie das Feld für Informationen verwenden, die sich bei jeder Bestellung ändern (z.B. gewünschter Liefertermin,..), wählen Sie die Checkbox „speichern“ nicht an! Die Daten des entsprechenden Feldes werden in diesem Fall zur Bestellung anstatt zum Kunden gespeichert.

 © 2001 R. Glanzmann & J. Fontanil



Layout Management

Im Layout-Management können Sie das Aussehen Ihres Shops komplett Ihren Bedürfnissen anpassen.

So ist es möglich, für jedes Frame (Fensterteil) eine eigene Hintergrundfarbe zu definieren oder ein Hintergrundbild anzeigen zu lassen.

Im Top-Frame (oberer Fensterteil) kann entweder der Shopname, ihr Shoplogo oder überhaupt nichts angezeigt werden. Den kleinen Stern, der neben dem „Warenkorb anzeigen"-Button (Knopf) eingeblendet wird, um in den Administrationsbereich zu gelangen, können Sie bei Bedarf ausblenden. ACHTUNG! Danach kommen Sie nur noch in den Administrationsbereich, indem Sie '`http://www.ihreshop.top_level_domain/shopname/shop/Admin/`' im Browser eingeben!

Beispiel: `http://www.pizzashop.de/pizzashop/shop/Admin/`

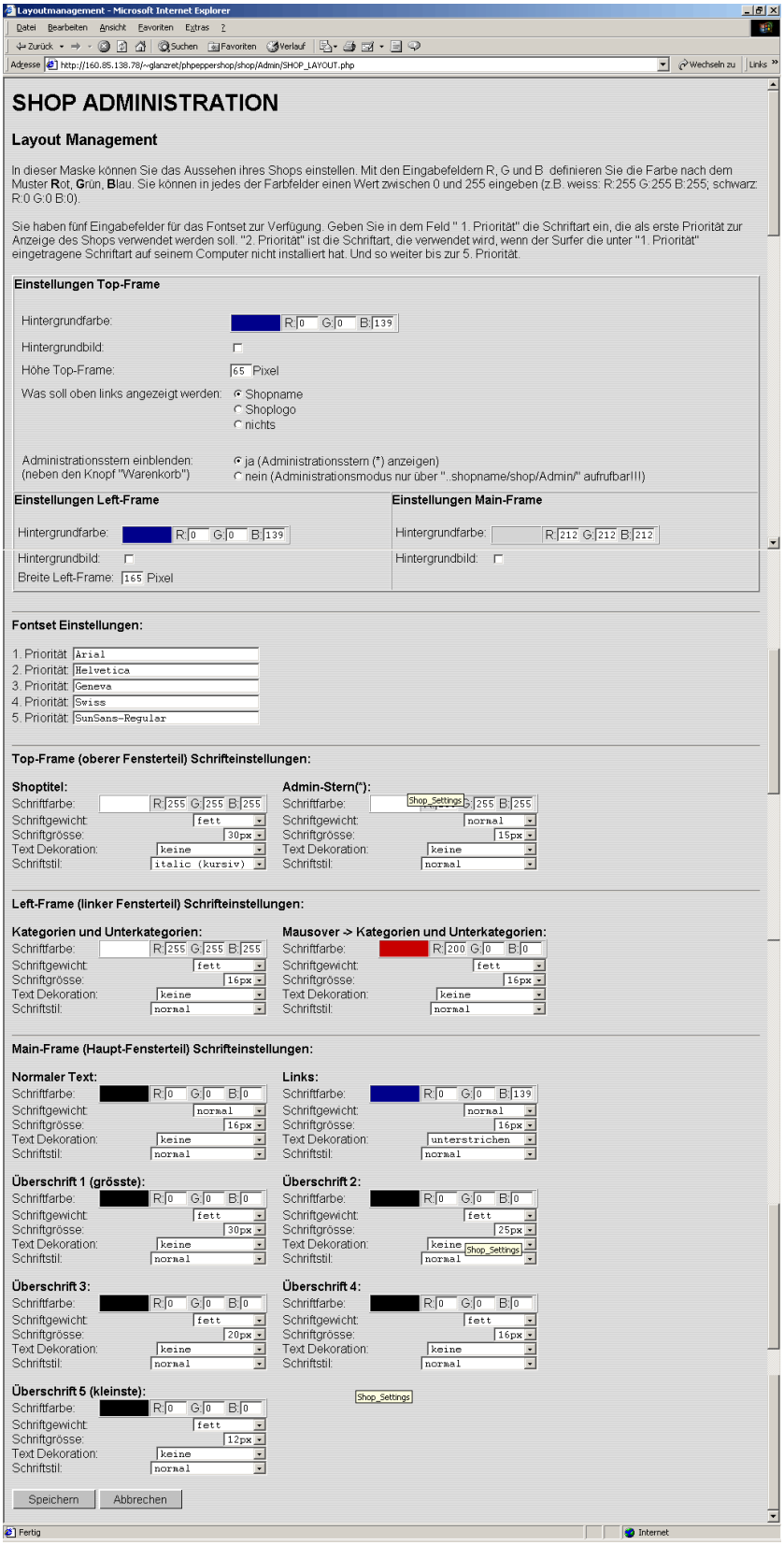
Sie haben fünf Eingabefelder für das Fontset zur Verfügung. Geben Sie in dem Feld '1. Priorität' die Schriftart ein, die mit höchster Priorität zur Anzeige des Shops verwendet werden soll. '2. Priorität' ist die Schriftart, die verwendet wird, wenn der Surfer die unter '1. Priorität' eingetragene Schriftart auf seinem Computer nicht installiert hat. So geht es weiter bis zur 5. Priorität.

 © 2001 R. Glanzmann & J. Fontanil

Die Schriftfarbe, Schriftgrösse, das Schriftgewicht, die Text-Dekoration und der Schriftstil können für alle verwendeten Tags (Schriftgruppen) gesondert eingestellt werden. Bitte beachten Sie, dass die Schriftstil-Einstellungen Browser-abhängig interpretiert werden.

Falls Sie die Grösse des Left-Frame (linker Fensterteil, welcher die Kategogien und Unterkategorien enthält) verändern wollen, können Sie das im Feld „Breite Left-Frame“ tun. Die Höhe des Top-Frames (oberer Fensterteil) kann im Feld „Höhe Top-Frame“ eingestellt werden.

 © 2001 R. Glanzmann & J. Fontanil



 © 2001 R. Glanzmann & J. Fontanil

Bilder (Hintergrund & Shoplogo) hochladen

Sie können für jedes Frame (Fensterteil) ein eigenes Hintergrundbild hochladen. Wenn Sie auf den Durchsuchen- bzw. Browse-Button klicken, öffnet sich ein Fenster, mit dem Sie eine Grafikdatei im Format GIF, JPG oder PNG auf Ihrer Festplatte auswählen können.

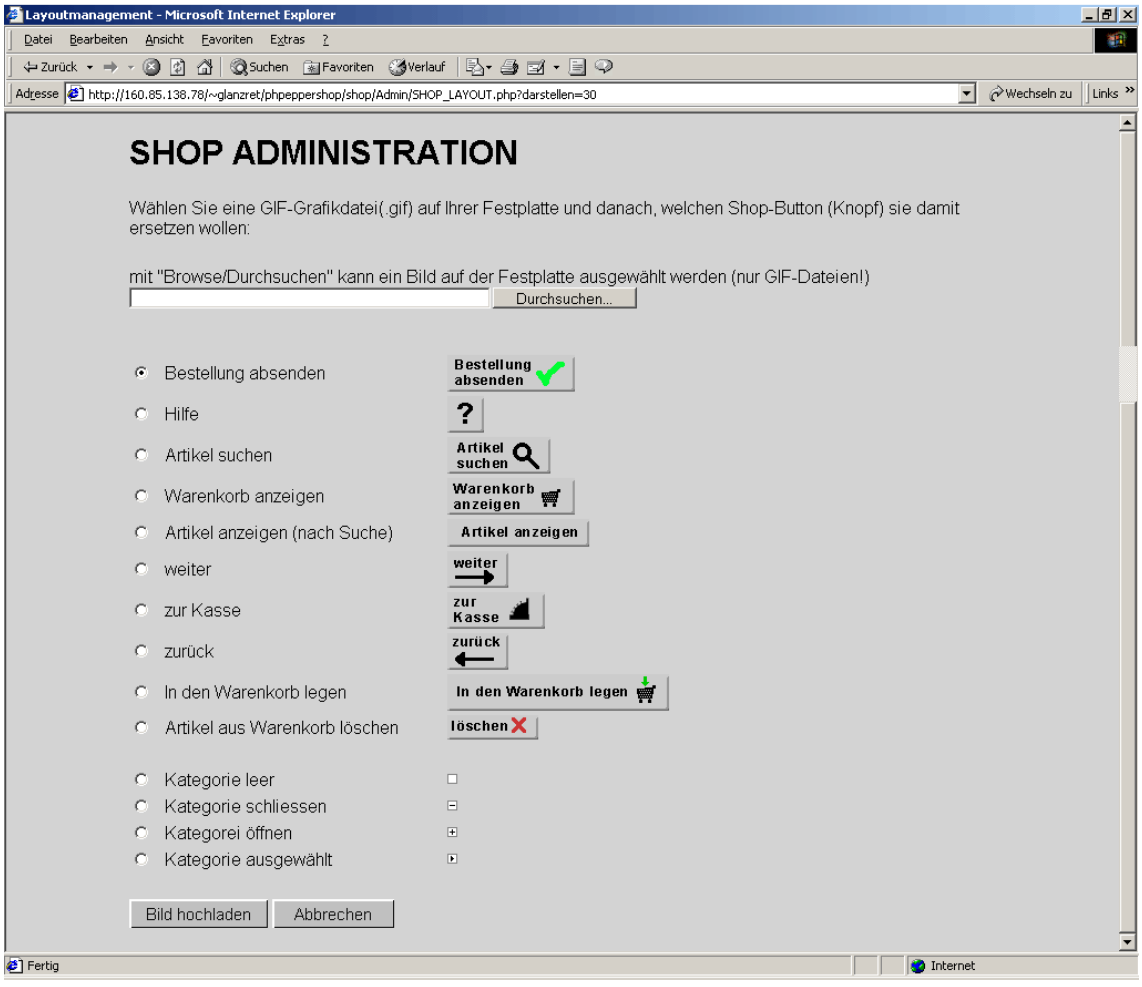
Danach müssen Sie den Verwendungszweck für das ausgewählte Bild bestimmen. Es kann entweder als Hintergrundbild für eines der drei Frames verwendet, oder als Shoplogo hochgeladen werden.

Sobald Sie ein Hintergrundbild oder ein Shoplogo hochgeladen haben, ist es aktiviert. Sollten Sie den Shop danach immer noch im alten Erscheinungsbild sehen, liegt das nur daran, dass die alte Version noch in ihrem Browsercache gespeichert ist.

Shopbuttons (Knöpfe hochladen)

Sie können jeden im Shop verwendeten Button durch einen eigenen Ihrer Wahl ersetzen. Die einzige Bedingung ist, dass der Button im GIF Format (erkennbar an der Dateiendung '.gif') abgespeichert wurde.

Wählen Sie, wie in 'Bilder (Hintergrund & Shoplogo) hochladen' beschrieben, eine Datei auf Ihrerer Festplatte aus. Wählen Sie dann, welcher Button damit ersetzt werden soll. Durch einen Klick auf 'Bild hochladen' wird der neue Button in den Shop hochgeladen.



 © 2001 R. Glanzmann & J. Fontanil

4. Anhang

4.1 Literaturverzeichnis

Nachfolgend eine Liste mit allen Quellen und Literaturverweisen, aufgegliedert nach den einzelnen Themen, die durch das Webshop-Programming tangiert werden.

4.1.1 PHP

PHP 4 Grundlagen und Profiwissen
Carl Hanser Verlag München Wien
Jörg Krause
November 2000
1161 Seiten
ISBN 3-446-21546-8

Dies war, wie auch in der Projektarbeit, unser 'Hauptbuch'. Es enthält eine Vielfalt an Themen, die alle ziemlich ausführlich erklärt werden. Aber auch dieses Buch hat seine Grenzen (Bilder in Datenbank speichern, grössere Projekte mit PHP)

PHP Grundlagen und Profiwissen
Carl Hanser Verlag München Wien
Jörg Krause
März 2000
1067 Seiten
ISBN 3-446-21301-5

Dieses Buch ist die Vorgänger-Version des oben genannten. Wir haben es selten konsultiert, aber doch ab und zu etwas daraus brauchen können.

PHP kurz & gut
O'Reilly Verlag Köln
Rasmus Lerdorf
1. Auflage 2000
111 Seiten

ISBN 3-89721-225-0

Eine praktische, kompakte Zusammenfassung, aber leider nur für PHP3.

PHP 4 Das bhv Taschenbuch
bhv Verlag Landsberg
Uwe Hess, Günther Karl
1. Auflage April 2001
764 Seiten
ISBN 3-8287-5057-5

Eine gute Einführung in die Grundlagen von PHP4, der Stoff geht aber nicht all zu tief.

Linux Enterprise 'PHP Special'-Ausgabe
Sonderausgabe Herbst 2001
98 Seiten

Ein Blick in die aktuelle PHP Entwicklung, mit Vorstellung der PHPLib und PEAR. Viele interessante, professionelle Ansätze.

PHP Archive / -Foren
http://www.hotscripts.com/
http://www.phpbuilder.com/
http://www.phparchiv.de/

Diese drei Pages halfen uns während der Entwicklung oft weiter. Viele Scripts findet man auf hotscripts und im phparchiv. Kompetente Foren in phpbuilder.

 © 2001 R. Glanzmann & J. Fontanil

4.1.2 Apache Webserver

Apache Web-Server Installation, Konfiguration, Administration
MITP-Verlag GmbH Bonn
Lars Eilebrecht
3. Auflage 2000
671 Seiten
ISBN 3-8266-0612-4

Ein sehr tief gehendes Buch. Half uns sehr gut bei der Apache-Konfiguration und beim Einrichten von `htaccess`.

4.1.3 CRM – Customer Relationship Management

iX Magazin für professionelle Informationstechnik

Ausgabe 8/2001, ab Seite 38
Operatives und analytisches CRM, CRM und Betriebswirtschaft

Ausgabe 7/2001, ab Seite 104
Dienen und Verdienen - CRM: Der Wert eines Kunden

Heinz Heise Verlag, 30604 Hannover

Wir haben umfassende und sehr aktuelle Artikel in der renommierten Zeitschrift iX gefunden.

Die Artikel geben dem Modewort CRM etwas Farbe. Man lernt, dass CRM nicht einfach eine Technik oder eine Sammlung von Funktionen ist, sondern eine Unternehmensphilosophie. Man muss, um ein umfangreiches CRM einzuführen, weit mehr machen, als es in einen Shop zu integrieren.

4.1.4 Datenbanken und im speziellen MySQL

MySQL in 21 Tagen
Markt + Technik Verlag München
Mark Maslakowski
2001
580 Seiten
ISBN 3-8272-5850-2

Dies war unser 'Hauptbuch' im Bereich Datenbanken. Man findet auch knifflige Aspekte von MySQL hier drin. Vor allem für die MySQL Konfiguration sehr zu empfehlen.

SQL, der Schlüssel zu relationalen Datenbanken
Rowohlt Taschenbuch Verlag GmbH Reineck bei Hamburg
Gregor Kuhlmann, Friedrich Müllmerstadt
Mai 1999
318 Seiten
ISBN 3-499-60063-3

Dieses Buch half uns vor allem bei schwierigen Queries, aber auch bei Fragen, wie man gewisse Datenbank-Aktionen überhaupt erst angehen soll. Sehr zu empfehlen.

Datenbanken – Vorlesung an der Zürcher Hochschule Winterthur
Dr. Hans-Walter Buff, zweites Jahr IT

Der Stoff half uns beim Entity Relationship Design, aber auch bei den Create-, Insert- und Del-SQL-Scripts.

 © 2001 R. Glanzmann & J. Fontanil

4.1.5 HTML, JavaScript und Cascading Style Sheets

HTML 4 – Die Sprache des Web
dpunkt-Verlag Heidelberg
Robert Tolksdorf
3. Auflage 1997
308 Seiten
ISBN 3-920993-77-2

Bietet einen tiefgehenden Einblick ins HTML, leider schon etwas in die Jahre gekommen.

Jetzt lerne ich HTML
Markt + Technik Verlag München
Harald Taglinger
1997
290 Seiten
ISBN 3-8272-5305-5

Gutes Nachschlagewerk (Farbtabelle, ...) immer wieder praktisch für 'wie ging das jetzt schon wieder?'-Momente. Leider auch schon etwas in die Jahre gekommen.

Das grosse Buch JavaScript
DATA BECKER GmbH & Co KG Düsseldorf
Rainer Kolbeck
1. Auflage 1997
422 Seiten
ISBN 3-8158-1321-2

Dieses Buch ist unbrauchbar (zumindest wenn man HTML schon kennt und den Netscape Javascript Debugger nicht erklärt braucht...)

JavaScript
verlag moderne industrie, 53227 Bonn
Michael Seeboerger-Weichselbaum
1. Auflage 2001
767 Seiten
ISBN 3-8266-8106-1

Wir konnten dieses Buch sehr gut als Nachschlagewerk gebrauchen, da es eine umfassende und logisch aufgebaute JavaScript Befehlsreferenz beinhaltet.

Wir können es nur weiter empfehlen, das Preis- / Leistungsverhältnis stimmt.

Einführung in JavaScript
http://atlas.ee.fhm.edu/mm-cd/skript/javascript/

Eine wirklich gelungene Einführung in JavaScript. Vermittelt die Grundlagen von JavaScript in nur 20 Minuten.

HTML-Referenz

http://www.teamone.de/selfhtml/

Selfhtml ist und bleibt das beste Nachschlagewerk in Sachen HTML. Sogar für JavaScript konnten wir einige Informationen von dieser Adresse gebrauchen. Sehr zu empfehlen.

4.1.6 Perl

Einführung in die Perl-Programmierung

http://userpage.fu-berlin.de/~corff/perl/perl-kurs.html

Dies ist ein guter Einstieg und trotzdem eine sehr umfangreiche Page. Wir haben vor allem das Kapitel 2, Grundlegende Elemente der Sprache, verwendet.

Perl – Ein 25 Minuten Crashkurs!

http://www.opensecure.de/PerlanB/perl_crashkurs.htm

Eine nette Einführung. Auf nur 20 Seiten hat man so ziemlich alle Sprachelemente, die man zu Beginn braucht, aufgelistet.

 © 2001 R. Glanzmann & J. Fontanil

Reguläre Ausdrücke in Perl http://www.oreilly.de/catalog/regexger/chapter/kap-2d.html http://www.heise.de/ix/artikel/1998/11/178/01.shtml	Details zur Muster Erkennung mit Regulären Ausdrücken mit Perl.

4.2 Software Versionen

4.2.1 Auf dem Server eingesetzte Software

SuSE Linux 7.1 Kernel: 2.4.0-4GB (i686)	Einzige Linux-Distribution, die zu Projektbeginn schon den Kernel 2.4.0 beinhaltete.
Apache Version 1.3.20 (Unix) (LAMPS Installation)	In SuSE Linux Distribution enthalten, genügend aktuell.
MySQL Version 3.23.35	Die zu Projektbeginn aktuellste MySQL Datenbank Version aus dem Internet.
PHP4 Version: 4.0.6	Diese damals aktuellste Version von PHP4 war glücklicherweise schon in der SuSE Distribution enthalten.
phpMyAdmin Version 2.2.0	Datenbank Administrationstool mit Webinterface. Dieses Programm brauchten wir, um SQL-Queries auszutesten und Änderungen in der Datenbank manuell vorzunehmen.

4.2.2 Auf den Arbeitsstationen eingesetzte Software

Windows 2000 Professionell SP1	Komfortable Entwicklungsoberfläche
Apache für Windows Version 1.3.19	Zu Testzwecken eingerichtet
PHP4 Version 4.04pl1	Zu Testzwecken eingerichtet
PHPed Version 2.96.1.2	Sehr nützlicher PHP-Editor mit integriertem FTP-Tool, welches direktes Arbeiten auf dem Server ermöglicht. Leider noch etwas Beta.
CUTE FTP Version 4.2.3 Build 2.14.1	FTP-Client
Putty Version 0.5.1	Telnet / SSH Client

 © 2001 R. Glanzmann & J. Fontanil

4.3 Projekttagebuch

Das Projekttagebuch wurde direkt während der Diplomarbeit jeweils am Ende jedes Arbeitstages weitergeführt. Es ist ziemlich detailliert, erhebt jedoch keinen Anspruch auf Vollständigkeit und soll vor allem den Ablauf des Projekts dokumentieren.

Mo, 10. September 01

- User pafei auf Webserver erstellt
- Drucker installiert
- PGP-installiert
- Analyse Feedbacks und E-Mails
- Erstellung ToDo-List mit Festlegung der Prioritäten und grober Aufwandabschätzung

Di, 11. September 01

- phpMyAdmin installiert
- htaccess für phpMyAdmin eingerichtet
- Account bei SourceForge eingerichtet und aktiviert (José Fontanil)
- Apache 1.3.20 installiert
- erfolglose Installation von PHP 4.0.6

Mi, 12. September 01

- Namen für Shop gesucht und gefunden: PhPepperShop
- Apache 1.3.20 konfiguriert (httpd.conf)
- Projekt PhPepperShop auf SourceForge eingereicht (geht ca. 72 Stunden bis Aktivierung)
- Von Hand installiert und konfiguriert:
  - PHP 4.0.6
  - OpenSSL 0.9.6b / ModSSL 2.8.4
  - Zlib-Library Version 1.1.3
  - IJG JPEG Software–Version 6b
  - Freetype Vers. 1.3.1
  - GD-Library 1.8.4
- Domain phpeppershop.com registriert (www.domainbank.net)
- Webserver für phpeppershop eingerichtet und aufgeschaltet (die Kosten werden momentan von Glarotech Informatikdienstleistungen getragen)

Do, 13. September 01

- Projektsitzung
- Projekt auf SourceForge eröffnet
- User glanzret auf SourceForge erstellt
- Einarbeitung in CVS
- PHP mit PDF-Unterstützung nochmals kompiliert

Fr, 14. September 01

- Internet von der ZHW die meiste Zeit nicht oder nur sehr langsam erreichbar -> viel warten!
- provisorische Website http://www.phpeppershop.com erstellt
- kleines Logo kreiert
- Mailinglisten auf SourceForge erstellt (devel, users, announce) und Test

 © 2001 R. Glanzmann & J. Fontanil

- Newsforum auf SourceForge eingerichtet und Test
- Homepage phpeppershop.sourceforge.net -> automatischer redirect auf www.phpeppershop.com

Mo, 17. September 01

- phpeppershop.com bei folgenden Suchmaschinen eingetragen: google.com, altavista.com, yahoo.de, search.ch
- Bericht „Webshop Framework“ im Internet Professionell 10/2001 (S.41)
- Einarbeitung in Perl
- Softlinks per Shell-Skript erstellen
- Script für Erstellung der Datenbank- und Connection-Skripts (1 oder 2 User, User erstellen oder schon erstellt, Shopname, Passwörter, User/Admin)
- Zeitplanung erstellt
- Einteilungen der Arbeiten nach Klassen
- 10'000 Hits auf der Webpage vom „Webshop Framework“ erreicht!

Di, 18. September 01

- Information für Leser der Zeitschrift „Internet Professionell“ auf Webserver veröffentlicht
- Verbindung zum Internet die meiste Zeit nicht möglich!!!
- Konzept Darstellung von Unterkategorien im linken Frame erstellt
- Kategorienmanagement Bedienkonzept entwickelt
- Fertigstellung Perl/Shell-Installationsscript (config.pl, config_part4.sh)

Mi, 19. September 01

- Kurze Einarbeitung Java Script (mouseover, down, out, gif)
- Java Skript Template erstellt, welches dann einfach in Shop integriert werden kann (dynamische Buttonsets werden damit möglich)
- Scrollen im Topframe unterbunden (CSS-File & Frameset)
- Charset 8859-1 im Mailversand ergänzt
- Expiring-Datum für Cache auf 1.1.1971 korrigiert
- PLZ nun auch alphanumerisch (USA) möglich
- LAMPS-Dokumentation erstellt
- Eingabesicherheit im Installationsscript erhöht
- Deinstallationsscript erstellt

Do, 20. September 01

- Konzept Layoutmanagement erstellt
- Dokumentation Installations/Deinstallationsscript
- yellowpay Anwendungskonzept angeschaut -> Implementation sollte kein Problem sein. Preis an Orbit abklären!
- Bugs in Installationsscript behoben
- Beginn mit Layoutmanagement
- Dokumentation „Einrichten eines Projekts auf SourceForge“

Fr, 21. September 01

- Dokumentation „Einrichten eines Projekts auf SourceForge“
- Layoutmanagement
- Parser für CSS-File-Handling entwickelt

 © 2001 R. Glanzmann & J. Fontanil

- CSS-Template mit Platzhalter-Tags erstellt

Mo, 24. September 01

- Artikel-Update Funktion neu programmiert
- CSS-Layout Management

Di, 25. September 01

- Gleicher Artikel mit verschiedenen Varianten und Optionen in Warenkorb
- CSS-Layout Management

Mi, 26. September 01

- Ganzer Tag Orbit/Comdex
- Besuch Case Study „Adori Webshop“

Do, 27. September 01

- CSS-Layout Management
- Anpassungen Installationsscript (MySQL-Fehler abfangen, Shopnamen Fehler)
- Unterkategorien implementieren

Fr, 28. September 01

- CSS-Layout Management
- Unterkategorien implementieren

Mo, 1. Oktober 01

- ZHW-LAN nicht verfügbar
- PC (Glanzmann) defekt
- CSS-Layout Management in Shop integrieren

Di, 2. Oktober 01

- Anpassung Layoutmanagement an Netscape 4.7
- Root-Problem gelöst!
- Unterkategorien fertig implementiert
- Beginn Artikel in mehrere Kategorien
- Artikelsuche um Unterkategorienfähigkeit und Mehrfachkategorien erweitert. Bilder können wahlweise angezeigt werden.

Mi, 3. Oktober 01

- Artikel in mehrere Kategorien
- Bugfixes
- Überarbeitung Admin-Layout und Anpassung Admin-CSS

Do, 4. Oktober 01

- Fertigstellung funktionaler Teil Mehrfachkategorien
- Fertigstellung darstellender Teil: Artikel in mehrere Kategorien

Fr, 5. Oktober 01

 © 2001 R. Glanzmann & J. Fontanil

- Fertigstellung Artikelhandling
- Kategorienmanagement Funktionalität
- Anpassung Artikelupdate-Funktion
- Delete-Funktion

Mo, 8. Oktober 01

- Möglichkeit zu wählen, wie viele Options und Variantenfelder zur Erfassung und Bearbeitung eines Artikels angezeigt werden (Mindestwert und Anzahl leere Felder)
- Shop-Administrationsmodus überarbeitet und der aktuellen Shopfunktionalität angepasst
- Eingabe-Plausibilitätsprüfung im Shop-Administrationsmodus per Java-Skript.
- Versandkostensettings

Di, 9. Oktober 01

- Versandkostensettings fertig (Formular muss nochmals leicht überarbeitet werden)
- Kategorienmanagement GUI (grosse Probleme Netscape 4.xx und Java-Script)

Mi, 10. Oktober 01

- Kategorienmanagement

Do, 11. Oktober 01

- Kategorienmanagement Sonderzeichen fähig gemacht
- Backup Shop angeschaut
- Kostenlose Artikel implementiert

Fr, 12. Oktober 01

- Datenbank auf CRM vorbereitet
- Analyse CRM und Erstellung CRM-Objekte
- Shop Sonderzeichen fähig gemacht

Mo, 15. Oktober 01

- CRM-Shop benutzerseitig implementiert

Di, 16. Oktober 01

- Dynamische Adresseingabefelder
- Suche erweitert um Anzahl Suchresultate
- Suche mit mehreren Begriffen möglich (wie bei www.google.com)

Mi, 17. Oktober 01

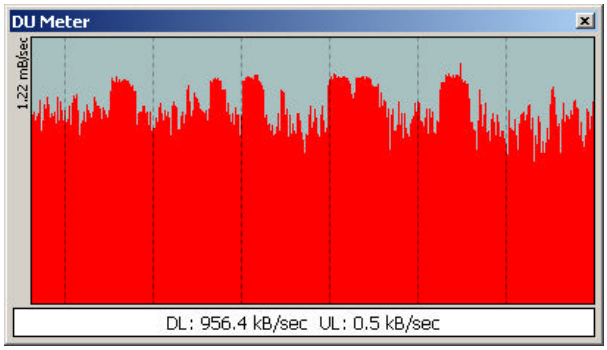
- GNU-PG Versuch 1
- Zahlungsmöglichkeiten / CRM
- Kreditkarten Datenbank , objektmässig, Administrator-mässig implementiert
- Versandkostensettings Update

Do, 18. Oktober 01

- Shopbuttons Upload fertiggestellt
- Nachnahmegebühr/Vorauskasse im Shop integriert.
- Update nicht möglich, da Netz nicht verfügbar

 © 2001 R. Glanzmann & J. Fontanil

- Ganzer Nachmittag Mailserver nicht verfügbar!
- Totalausfall LAN, nichts geht mehr! Stundenlang!



- Kreditkarten Zahlungsworkflow-Analyse
- GNU-PG Versuch 2
- Artikelhandling überarbeitet

Fr, 19. Oktober 01

- Eigene Pop-Up und Hilfe-Funktionen für Administrationsmodus, die mit dem CSS-File des Administrationsmodus arbeiten.
- Kreditkarten Zahlungsworkflow beendet (Sa.)
- Buttons-Upload um Kategoriebuttons erweitert
- Admin-Buttons einheitlich erstellt (Sa.)
- Testergebnisse abgearbeitet und Bugs behoben (Sa.)
- Bestellmanagement an CRM anpassen
- Raumreservierung für Präsentationen DA (½ Tag)

Mo, 22. Oktober 01

- Festlegung Layout für Dokumentation
- Beginn Dokumentation

Di, 23. Oktober 01 – Fr, 26 Oktober

- Dokumentation Diplomarbeit

Mo, 29. Oktober 01

- Abgabe Diplomarbeit

 © 2001 R. Glanzmann & J. Fontanil

4.4 GPL – GNU General Public Licence

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation herausgegeben, weitergeben und/oder modifizieren, entweder unter Version 2 der Lizenz oder (wenn Sie es wünschen) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE JEDE GEWÄHRLEISTUNG – sogar ohne die implizite Gewährleistung der MARKTREIFE oder der EIGNUNG FÜR EINEN BESTIMMTEN ZWECK.

Details finden Sie in der GNU General Public License. Sie sollten eine Kopie der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, schreiben Sie an die Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

4.4.1 Deutsche Fassung der GPL (Auszug)

Übersetzt von Katja Lachmann Übersetzungen *(na194@fim.uni-erlangen.de*), modifiziert von Peter Gerwinski *(peter.gerwinski@uni-essen.de*) und Jörg Krause *(joerg@krause.net*). Es handelt sich nicht um eine offizielle oder im rechtlichen Sinne anerkannte Übersetzung.
Die Free Software Foundation (FSF) ist nicht der Herausgeber dieser Übersetzung, und sie hat diese Übersetzung auch nicht als rechtskräftigen Ersatz für die Original GNU-GPL anerkannt. Da die Übersetzung nicht sorgfältig von Anwälten über-prüft wurde, können die Übersetzer nicht garantieren, dass die Übersetzung die rechtlichen Aussagen der GNU-GPL exakt wiedergibt. Wenn Sie sichergehen wollen, dass von Ihnen geplante Aktivitäten im Sinne der GNU-GPL gestattet sind, halten Sie sich bitte an die englischsprachige Originalversion. Die Free Software Foundation möchte Sie darum bitten, diese Übersetzung nicht als offizielle Lizenzbedingungen für von Ihnen geschriebene Programme zu verwenden. Bitte benutzen Sie hierfür stattdessen die von der Free Software Foundation herausgegebene englischsprachige Originalversion. Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung

§0 Diese Lizenz gilt für jedes Programm und jedes andere Werk, in dem ein entsprechender Vermerk des Copyright-Inhabers darauf hinweist, dass das Werk unter den Bestimmungen dieser General Public License verbreitet werden darf. Im Folgenden wird jedes derartige Programm oder Werk als »das Programm «bezeichnet; die Formulierung »auf dem Programm basierendes Werk« bezeichnet das Programm sowie jegliche Bearbeitung des Programms im urheberrechtlichen Sinne, also ein Werk, welches das Programm, auch auszugsweise, sei es unverändert oder verändert oder in eine andere Sprache übersetzt, enthält. Andere Handlungen als Vervielfältigung, Verbreitung und Bearbeitung werden von dieser Lizenz nicht berührt; sie fallen nicht in ihren Anwendungsbereich. Der Vorgang der Ausführung des Programms wird nicht eingeschränkt und die Ausgaben des Programms unterliegen dieser Lizenz nur, wenn der Inhalt ein auf dem Programm basierendes Werk darstellt (unabhängig davon, dass die Ausgabe durch die Ausführung des Programmes erfolgte). Ob dies zutrifft, hängt von den Funktionen des Programms ab.

§1 Sie dürfen auf beliebigen Medien unveränderte Kopien des Quelltextes des Programms, wie sie ihn erhalten haben, anfertigen und verbreiten. Voraussetzung hierfür ist, dass Sie mit jeder Kopie einen entsprechenden Copyright-Vermerk sowie einen Haftungsausschluss veröffentlichen, alle Vermerke, die sich auf diese Lizenz und das Fehlen einer Garantie beziehen, unverändert lassen und des Weiteren allen anderen Empfängern des Programms zusammen mit dem Programm eine Kopie dieser Lizenz zukommen lassen. Sie dürfen für den eigentlichen Kopiervorgang eine Gebühr verlangen. Wenn Sie es wünschen, dürfen Sie auch gegen Entgelt eine Garantie für das Programm anbieten.

§2 Sie dürfen Ihre Kopien des Programms oder eines Teils davon verändern, wodurch ein auf dem Programm basierendes Werk entsteht; Sie dürfen derartige Bearbeitungen unter den Bestimmungen von Paragraph 1 vervielfältigen und verbreiten, vorausgesetzt, dass zusätzlich alle folgenden Bedingungen er-füllt werden:

(a) Sie müssen die veränderten Dateien mit einem auffälligen Vermerk versehen, der auf die von Ihnen vorgenommene Modifizierung und das Datum jeder Änderung hinweist.
(b) Sie müssen dafür sorgen, dass jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von dem Programm oder Teilen davon abgeleitet ist, Dritten gegenüber als Ganzes unter den Bedingungen dieser Lizenz ohne Lizenzgebühren zur Verfügung gestellt wird.
(c) Wenn das veränderte Programm normalerweise bei der Ausführung inter-aktiv Kommandos einliest, müssen Sie dafür sorgen, dass es, wenn es auf dem üblichsten Wege für solche interaktive Nutzung gestartet wird, eine Meldung ausgibt oder ausdruckt, die einen geeigneten Copyright-Vermerk enthält sowie einen Hinweis, dass es keine Gewährleistung gibt (oder an-derenfalls, dass Sie Garantie leisten), und dass die Benutzer das Programm unter diesen Bedingungen weiter verbreiten dürfen. Auch muss der Benutzer darauf hingewiesen werden, wie er eine Kopie dieser Lizenz ansehen kann. (Ausnahme: Wenn das Programm selbst interaktiv arbeitet, aber normalerweise keine derartige Meldung ausgibt, muss Ihr auf dem Programm basierendes Werk auch keine solche Meldung ausgeben.) Diese Anforderungen betreffen das veränderte Werk als Ganzes. Wenn identifizierbare Abschnitte des Werkes nicht von dem Programm abgeleitet sind und vernünftigerweise selbst als unabhängige und eigenständige Werke betrachtet werden können, dann erstrecken sich diese Lizenz und ihre Bedingungen nicht auf diese Abschnitte, wenn sie als eigenständige Werke verbreitet werden. Wenn Sie jedoch dieselben Abschnitte als Teil eines Ganzen

 © 2001 R. Glanzmann & J. Fontanil

verbreiten, das ein auf dem Programm basierendes Werk darstellt, dann muss die Verbreitung des Ganzen nach den Bedingungen dieser Lizenz erfolgen, deren Bedingungen für weitere Lizenznehmer somit auf die Gesamtheit aus-gedehnt werden – und damit auf jeden einzelnen Teil, unabhängig vom jeweiligen Autor. Somit ist es nicht die Absicht dieses Abschnittes, Rechte für Werke in Anspruch zu nehmen oder zu beschneiden, die komplett von Ihnen geschrieben wurden; vielmehr ist es die Absicht, die Rechte zur Kontrolle der Verbreitung von Werken, die auf dem Programm basieren oder unter seiner auszugsweisen Verwendung zusammengestellt worden sind, auszuüben. Ferner bringt ein einfaches Zusammenstellen eines anderen Werkes, das nicht auf dem Programm basiert, zusammen mit dem Programm oder einem auf dem Programm basierenden Werk auf ein- und demselben Speicher- oder Bertriebsmedium das andere Werk nicht in den Anwendungsbereich dieser Lizenz.

§3 Sie dürfen das Programm (oder ein darauf basierendes Werk gemäß Paragraph 2) als Objectcode oder in ausführbarer Form unter den Bedingungen von Paragraph 1 und 2 vervielfältigen und verbreiten – vorausgesetzt, dass Sie außerdem eine der folgenden Leistungen erbringen:
(a) Liefern Sie das Programm zusammen mit dem vollständigen zugehörigen maschinenlesbaren Quelltext auf einem für den Datenaustausch üblichen Medium aus, wobei die Verteilung unter den Bedingungen der Paragraphen 1 und 2 erfolgen muss. Oder:
(b) Liefern Sie das Programm zusammen mit einem mindestens drei Jahre lang gültigen schriftlichen Angebot aus, jedem Dritten eine vollständige maschinenlesbare Kopie des Quelltextes zur Verfügung zu stellen – zu nicht höheren Kosten als denen, die durch den physikalischen Kopiervorgang anfallen –, wobei der Quelltext unter den Bedingungen der Paragraphen 1 und 2 auf einem für den Datenaustausch üblichen Medium weiter-gegeben wird. Oder:
(c) Liefern Sie das Programm zusammen mit dem schriftlichen Angebot der Zurverfügungstellung des Quelltextes aus, das Sie selbst erhalten haben. (Diese Alternative ist nur für nicht-kommerzielle Verbreitung zulässig und nur, wenn Sie das Programm als Objectcode oder in ausführbarer Form mit einem entsprechenden Angebot gemäß Absatz b erhalten haben.) Unter dem Quelltext eines Werkes wird diejenige Form des Werkes verstanden, die für Bearbeitungen vorzugsweise verwendet wird. Für ein ausführbares Programm bedeutet »der komplette Quelltext«: Der Quelltext aller im Programm enthaltenen Module einschließlich aller zugehörigen Modulschnittstellen-Definitionsdateien sowie der zur Compilation und Installation verwendeten Skripte. Als besondere Ausnahme jedoch braucht der verteilte Quelltext nichts von dem zu enthalten, was üblicherweise (entweder als Quelltext oder in binärer Form) zusammen mit den Hauptkomponenten des Betriebssystems (Kernel, Compiler usw.) geliefert wird, unter dem das Programm läuft – es sei denn, diese Komponente selbst gehört zum ausführbaren Programm. Wenn die Verbreitung eines ausführbaren Programms oder des Objectcodes dadurch erfolgt, dass der Kopierzugriff auf eine dafür vorgesehene Stelle gewährt wird, so gilt die Gewährung eines gleichwertigen Zugriffs auf den Quelltext als Verbreitung des Quelltextes, auch wenn Dritte nicht dazu gezwungen sind, den Quelltext zusammen mit dem Objectcode zu kopieren.

§4 Sie dürfen das Programm nicht vervielfältigen, verändern, weiter lizenzieren oder verbreiten, sofern es nicht durch diese Lizenz ausdrücklich gestattet ist. Jeder anderweitige Versuch der Vervielfältigung, Modifizierung, Weiterlizenzierung und Verbreitung ist nichtig und beendet automatisch Ihre Rechte unter dieser Lizenz. Jedoch werden die Lizenzen Dritter, die von Ihnen Kopien oder Rechte unter dieser Lizenz erhalten haben, nicht beendet, solange diese die Lizenz voll anerkennen und befolgen.

§5 Sie sind nicht verpflichtet, diese Lizenz anzunehmen, da Sie sie nicht unter-zeichnet haben. Jedoch gibt Ihnen nichts anderes die Erlaubnis, das Programm oder von ihm abgeleitete Werke zu verändern oder zu verbreiten. Diese Handlungen sind gesetzlich verboten, wenn Sie diese Lizenz nicht anerkennen. Indem Sie das Programm (oder ein darauf basierendes Werk) verändern oder verbreiten, erklären Sie Ihr Einverständnis mit dieser Lizenz und mit allen ihren Bedingungen bezüglich der Vervielfältigung, Verbreitung und Veränderung des Programms oder eines darauf basierenden Werkes.

§6 Jedesmal, wenn Sie das Programm (oder ein auf dem Programm basierendes Werk) weitergeben, erhält der Empfänger automatisch vom ursprünglichen Lizenzgeber die Lizenz, das Programm entsprechend den hier festgelegten Bestimmungen zu vervielfältigen, zu verbreiten und zu verändern. Sie dürfen keine weiteren Einschränkungen der Durchsetzung der hierin zugestandenen Rechte des Empfängers vornehmen. Sie sind nicht dafür verantwortlich, die Einhaltung dieser Lizenz durch Dritte durchzusetzen.

§7 Sollten Ihnen infolge eines Gerichtsurteils, des Vorwurfs einer Patentverlet-zung oder aus einem anderen Grunde (nicht auf Patentfragen begrenzt) Be-.Copyright und Lizenzhinweise 9 Bedingungen (durch Gerichtsbeschluss, Vergleich oder anderweitig) auferlegt werden, die den Bedingungen dieser Lizenz widersprechen, so befreien Sie diese Umstände nicht von den Bestimmungen dieser Lizenz. Wenn es Ihnen nicht möglich ist, das Programm unter gleichzeitiger Beachtung der Bedingungen in dieser Lizenz und Ihrer anderweitigen Verpflichtungen zu verbreiten, dann dürfen Sie als Folge das Programm überhaupt nicht verbreiten. Wenn zum Beispiel ein Patent nicht die gebührenfreie Weiterverbreitung des Programms durch diejenigen erlaubt, die das Programm direkt oder indirekt von Ihnen erhalten haben, dann besteht der einzige Weg, sowohl das Patentrecht als auch diese Lizenz zu befolgen, darin, ganz auf die Verbreitung des Programms zu verzichten. Sollte sich ein Teil dieses Paragraphen als ungültig oder unter bestimmten Umständen nicht durchsetzbar erweisen, so soll dieser Paragraph seinem Sin-ne ach angewandt werden; im Übrigen soll dieser Paragraph als Ganzes gelten. Zweck dieses Paragraphen ist nicht, Sie dazu zu bringen, irgendwelche Patente oder andere Eigentumsansprüche zu verletzen oder die Gültigkeit solcher Ansprüche zu bestreiten; dieser Paragraph hat einzig den Zweck, die Integrität des Verbreitungssystems der freien Software zu schützen, das durch die Praxis öffentlicher Lizenzen verwirklicht wird. Viele Leute haben großzügige Beiträge zu dem großen Angebot der mit diesem System verbreiteten Software im Vertrauen auf die konsistente Anwendung dieses Systems geleistet; es liegt am Autor/Geber, zu entscheiden, ob er die Software mittels irgendeines anderen Systems verbreiten will; ein Lizenznehmer hat auf diese Entscheidung keinen Einfluss. Dieser Paragraph ist dazu gedacht, deutlich klarzustellen, was als Konsequenz aus dem Rest dieser Lizenz betrachtet wird.

§8 Wenn die Verbreitung oder die Benutzung des Programms in bestimmten Staaten entweder durch Patente oder durch urhe-

 © 2001 R. Glanzmann & J. Fontanil

berrechtlich geschützte Schnittstellen eingeschränkt ist, kann der Urheberrechtsinhaber, der das Programm unter diese Lizenz gestellt hat, eine explizite geographische Begrenzung der Verbreitung angeben, in der diese Staaten ausgeschlossen werden, so dass die Verbreitung nur innerhalb und zwischen den Staaten erlaubt ist, die nicht ausgeschlossen sind. In einem solchen Fall beinhaltet diese Lizenz die Beschränkung, als wäre sie in diesem Text niedergeschrieben.

§9 Die Free Software Foundation kann von Zeit zu Zeit überarbeitete und/oder neue Versionen der General Public License veröffentlichen. Solche neuen Versionen werden vom Grundprinzip her der gegenwärtigen entsprechen, können aber im Detail abweichen, um neuen Problemen und Anforderungen gerecht zu werden. Jede Version dieser Lizenz hat eine eindeutige Versionsnummer. Wenn in einem Programm angegeben wird, dass es dieser Lizenz in einer bestimmten Versionsnummer oder »jeder späteren Version« unterliegt, so haben Sie die Wahl, entweder den Bestimmungen der genannten Version zu folgen oder denen jeder beliebigen späteren Version, die von der Free Software Foundation veröffentlicht wurde. Wenn das Programm keine Versionsnummer angibt, können Sie eine beliebige Version wählen, die je von der Free Software Foundation veröffentlicht wurde.

§10 Wenn Sie den Wunsch haben, Teile des Programms in anderen freien Programmen zu verwenden, deren Bedingungen für die Verbreitung anders sind, schreiben Sie an den Autor, um ihn um die Erlaubnis zu bitten. Für Software, die unter dem Copyright der Free Software Foundation steht, schreiben Sie an die Free Software Foundation; wir machen zu diesem Zweck gelegentlich Ausnahmen. Unsere Entscheidung wird von den beiden Zielen geleitet wer-den, zum einen den freien Status aller von unserer freien Software abgeleiteten Werke zu erhalten und zum anderen das gemeinschaftliche Nutzen und Wiederverwenden von Software im Allgemeinen zu fördern.

§11 Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Copyright-Inhaber und/oder Dritte das Programm so zur Verfügung, »wie es ist«, ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich – aber nicht begrenzt auf – Marktreife oder Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms liegt bei Ihnen. Sollte sich das Programm als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.

§12 In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Copyright-Inhaber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher allgemeiner oder spezieller Schäden, Schäden durch Seiteneffekte (Nebenwirkungen) oder Folgeschäden, die aus der Benutzung des Programms oder der Unbenutzbarkeit des Programms folgen (einschließlich – aber nicht beschränkt auf – Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder anderen getragen werden müssen, oder dem Unvermögen des Programms, mit irgendeinem anderen Programm zusammenzuarbeiten), selbst wenn ein Copyright-Inhaber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.

 © 2001 R. Glanzmann & J. Fontanil

4.5 Detailliertes Inhaltsverzeichnis

1. Bericht

1.1 Kurzbeschrieb in Deutsch und Englisch....3
1.1.1 Kurzbeschrieb (Deutsch)....3
1.1.2 Abstract (English)....3
1.2 Vorwort....5
1.3 Aufgabenstellung....6
Umfeld....6
Aufgaben....6
Aufgabendetails....6
Termine....7
Kontaktmöglichkeiten....7
Abgabe....7
Bewertung....7
1.4 Analyse des erhaltenen Feedbacks....8
1.4.1 E-Mail Auswertung....8
Weitere Tipps aus den E-Mails....8
1.4.2 Auflistung aller erhaltenen Feedbacks....9
1.4.3 Erste Vorentscheidungen....10
1.5 Wahl der verwendeten Technologien....12
1.6 Zeitplanung....14
1.6.1 Aufgabenauflistung nach Wichtigkeit, 17.09.2001....14
1.6.2 Zeitplanung, erstellt zu Projektbeginn....16
1.6.3 Effektiver Ablauf des Projekts....16
1.7 Tests....18
1.7.1 Usability-Test....18
1.7.2 Extremwerte Test....20
1.7.3 Funktionstest....21
Test des CRM's (Customer Relationship Management)....22
Test des Bestellungsablaufs....23
1.8 Erfahrungen....24
1.8.1 Installation der Systemumgebung - LAMPS....24
Inhalt dieses Kapitels....24
Wieso diese Kombination?....24
Installation....25
Quellen beschaffen....25
Linux....25
Apache....25
MySQL....25
PHP....25
SSL....25
Erweiterungen....25
Eigentliche Installation....25
Entpacken der Pakete....25
MySQL....26
SSL Installation....26
Apache und mod_SSL....26
Server-Zertifikat erstellen....27
Apache Installation vervollständigen....28

 © 2001 R. Glanzmann & J. Fontanil

PHP....28
Freetype....28
PDFlib....28
Zlib....28
IJG JPEG Software....29
GD....29
PHP konfigurieren und kompilieren....29
Konfiguration....30
Starten....30
1.8.2 Einrichten eines Passwort geschützten Bereichs mit htaccess....31
1.8.3 SourceForge einrichten....33
Einleitung....33
SourceForge einrichten....33
Eigener Account....33
Ein neues Projekt lancieren....33
Das neue Projekt einrichten....33
Übersicht des Projekts....34
Trove Categorisation....34
Public Information....34
Entwickler einrichten....35
Webseiten für das Projekt aufsetzen....35
Mailinglisten anlegen....35
News-Foren....36
Dokumentation....36
(Bug-, Support-, Patch-, Feature-Request-) Tracker....36
CVS einrichten....37
Was ist CVS?....37
CVS Entwicklungszyklus....37
CVS für sein Projekt auf SourceForge einrichten....37
Den Quellcode ins Repository importieren (Import)....38
Arbeitskopie aus dem Repository auschecken (Checkout)....38
Upgedatete Datei zurück ins Repository schreiben (Update, Commit)....39
Eine neue Datei ins Repository einfügen (Add, Commit)....39
Webzugriff auf das Repository (Diff, ...)....39
Eine Datei oder ein Verzeichnis entfernen....40
Eine Datei verschieben....40
CVS Id-Tag....40
Weitere Services von SourceForge....41
Aufgaben (Task Management)....41
Surveys....41
Compile Farm....41
MySQL Datenbank....41
Virtual Hosts Service....41
Fragen zu SourceForge....41
1.8.4 Browser Kompatibilität....42
1.8.5 PHP und Safe-Mode....43
1.8.6 GNU-PG....44
1.8.7 Strukturen beim Webprogramming....46
Variablentransport zwischen Funktionen....46
Andere Fehlermeldungen....47

 © 2001 R. Glanzmann & J. Fontanil

Interaktivität einer HTML-Seite....48
Entwurf von neuen Komponenten für den PhPepperShop: (Entwicklungszyklus)....49
1.8.8 Variablen zwischen den Webpages transportieren....51
Übermittlung direkt im Link (GET)....51
Übermittlung als Feld in einem Formular (POST oder GET)....51
Welche Daten sollen wie übergeben werden?....52
PHP-Funktionen: addslashes und stripslashes....53
PHP-Funktionen: urlencode und urldecode....54
PHP-Funktion: htmlspecialchars....54
1.8.9 JavaScript....55
Gleichzeitige Aktualisierung von Frames mittels JavaScript....55
Überprüfung eines Formularfeldes auf Eingabe....56
Unsere Meinung zu JavaScript....57
1.8.10 Datenbankdesign....58
Unterkategorien....58
Shop-Settings....58
1.9 Zukunft des PhPepperShops....60
1.9.1 Veröffentlichung des Final-Release....60
1.9.2 Leute, die den Shop weiterentwickeln wollen....60
1.10 Schlusswort....62
2. Architektur
2.1 Security....63
2.1.1 PHP....63
2.1.2 htaccess....63
2.1.3 MySQL....63
2.1.4 SSL....64
2.1.5 GNU-PG....64
2.1.6 Kunden-ID....65
2.2 PHP-Module und -Funktionen....66
2.2.1 Übersicht....66
2.2.2 PHP-Module und ihre Funktionen....68
Klassendefinitionen....68
2.2.3 Fehlerbehandlung in den Modulen....98
Allgemeine Sicherheitsvorkehrungen....98
Fehlercode....98
Beispiele....98
2.3 Session Management....100
2.3.1 Maximale Session Dauer....100
2.3.2 Session-Management-Header....100
2.4 Datenbank....102
2.4.1 Idee und Konzept....102
2.4.2 Personalisierung durch das Installationstool config.pl....102
2.4.3 Relationen in der Datenbank (Übersicht)....103
2.4.4 Entity Relationship Diagramm der Shop Datenbank....103
ER Diagramm Teil 1....104
ER Diagramm Teil 2....105
ER Diagramm Teil 3....106
2.4.5 Tabellen und ihre Attribute im Detail....107
2.4.6 SQL-Query Handling....109
2.4.7 DB-Wrapper (database.php)....110

 © 2001 R. Glanzmann & J. Fontanil

2.5 Shop Layout Management....111
2.5.1 Shop Layout Beschreibung....111
2.5.2 Layout Administrationsmodus....111
2.5.3 Layout Management....112
Layout Management....112
Hintergrundbilder und Shoplogo hochladen....113
Buttons hochladen....113
2.5.4 Buttons im Administrationsbereich....115
2.6 Artikelmanagement....117
2.6.1 Artikelsuche....119
Mehrere Suchbegriffe in einer Abfrage erlaubt....119
Bilder können optional angezeigt werden....119
Mehrfachkategorien Anzeige....119
Inkrementangabe....119
Screenshot....120
2.7 Kategorienmanagement....121
2.7.1 Ursprüngliches Konzept (unverändert)....121
Kategorienmanagement Bildschirmansicht:....121
neue Kategorie erstellen [1]:....122
Kategorie verschieben [2]:....122
Kategorie umbenennen [3]:....122
Kategorie löschen [4]:....122
neue Unterkategorie erstellen [5]:....122
Unterkategorie innterhalb Kategorie verschieben [6]:....123
Unterkategorie in eine andere Kategorie verschieben [7]:....123
Unterkategorie umbenennen [8]:....123
Unterkategorie löschen [9]:....123
2.7.2 Hauptmenü....124
2.8 Zahlungsmöglichkeiten....125
2.8.1 Vorauskasse, Rechnung, Nachnahme und kostenlos....125
2.8.2 Kreditkarten....125
Shop-intern abgewickelte Kreditkartenzahlung....125
Einschränkungen....125
Shop-extern abgewickelte Kreditkartenzahlung....125
Idee und Konzept....126
Usability der Schnittstelle....126
Das Testmodul (pay_ext_test.php)....127
2.9 Bestellungsmanagement....128
2.10 Kundenattribute Management....129
2.11 CRM – Customer Relationship Management....130
2.11.1 Welche Funktionen bietet das CRM....130
3. Manuals
3.1 Installation / Deinstallation....132
3.1.1 Automatisierte Installation....132
3.1.2 Ablauf....132
3.1.3 Installationsmechanismus....133
3.1.4 Erste Konfigurationen....134
3.1.5 Deinstallation des Shops....134
3.2 Backup und Restore mit MySQL....135
3.2.1 Automatisiertes Datenbank-Backup....135

 © 2001 R. Glanzmann & J. Fontanil

Dateinamen....136
phpMyBackup....136
3.2.2 Restore eines Datenbank-Backups....137
3.2.3 Backup der Shopdateien und Shopbilder von Hand....137
3.2.4 Backup der Shop Datenbank von Hand....137
3.2.5 Restore von Hand....138
3.3 Bedienung....139
3.3.1 Bedienung des Shops....139
3.3.2 Bedienung des Administrationstools....139
Allgemeine Shopeinstellung....139
Versandkosten Einstellungen....143
Kundenattribute bearbeiten....144
Layout Management....145
Bilder (Hintergrund & Shoplogo) hochladen....148
Shopbuttons (Knöpfe hochladen)....148
4. Anhang
4.1 Literaturverzeichnis....149
4.1.1 PHP....149
4.1.2 Apache Webserver....149
4.1.3 CRM – Customer Relationship Management....150
4.1.4 Datenbanken und im speziellen MySQL....150
4.1.5 HTML, JavaScript und Cascading Style Sheets....150
4.1.6 Perl....151
4.2 Software Versionen....152
4.2.1 Auf dem Server eingesetzte Software....152
4.2.2 Auf den Arbeitsstationen eingesetzte Software....152
4.3 Projekttagebuch....153
4.4 GPL – GNU General Public Licence....158
4.4.1 Deutsche Fassung der GPL (Auszug)....158
4.5 Detailliertes Inhaltsverzeichnis....161
4.6 Inhalt der CD....166
4.7 Zusammenfassungen....167
4.8 Source Code....168

 © 2001 R. Glanzmann & J. Fontanil

4.6 Inhalt der beiliegenden CD

Dokumentation

- Open-Source-Manuals
- Zusammenfassungen-Notizen

Webshop

- Button Templates Administratiosbereich
- PhPepperShop
- Source
  - shop
    - Admin
    - Bilder
    - Buttons
    - Frameset
    - template

 © 2001 R. Glanzmann & J. Fontanil

4.7 Zusammenfassungen

Auf der CD finden Sie folgende Zusammenfassungen und Anleitungen, die wir während der Projekt- und Diplomarbeit für unseren Gebrauch erstellt haben. Diese gehören eigentlich nicht zum Diplomarbeitsbericht. Wir denken aber, dass sie jemandem, der den Shop installieren oder weiterentwickeln will, sehr nützlich sind.

- apache_konfigurieren.sdw
- cvs.html
- Datenbankdesign - Relationship - word.doc
- DB-Wrapper.doc
- Dynamische Variablen in PHP.doc
- gla-linux-hilfe.sdw
- htaccess einrichten.sdw
- HTML Befehlszusammenfassung.doc
- mailversand_mit_php4.sdw
- MySQL Infos.doc
- mysql-notizen.sdw
- mysql-sicherheit.sdw
- MySQL_INSTALL_BINARY.doc
- notizen_php.sdw
- PHP4 Referenz.doc
- prg_bsp_filelesenschreiben.sdw

 © 2001 R. Glanzmann & J. Fontanil

4.8 Source Code

Der Source Code des PhPepperShops befindet sich im zweiten Buch. Die Dokumente sind in folgender Reihenfolge sortiert abgelegt worden. Der Quellcode ist selbstverständlich auch auf der CD gespeichert.

Installationstool

config.pl
config_part4.pl

Redirect-Page und Frameset

index.html
index_php.pdf

User Level Scripts

artikel_def.php
attribut_def.php
bestellung_def.php
bild_view.php
database.php
initialize.php
kategorie_def.php
kreditkarte_def.php
kunde_def.php
payment_interface.php
pay_def.php
pay_ext_test.php
pop_up.php
USER_ADMIN_HILFE.php
USER_ARTIKEL_HANDLING.php
USER_ARTIKEL_HANDLING_AUFRUF.php
USER_BESTELLUNG.php
USER_BESTELLUNG_1.php
USER_BESTELLUNG_AUFRUF.php
USER_BESTELLUNG_DARSTELLUNG.php
USER_SQL_BEFEHLE.php
versandkosten_def.php

Admin Level Scripts

ADMIN_backup.php
ADMIN_HILFE.php
ADMIN_initialize.php
ADMIN_restore.php
ADMIN_SQL_BEFEHLE.php
bild_up.php
csstemplate.txt
demo_htaccess.php

 © 2001 R. Glanzmann & J. Fontanil

index_admin.php
indextemplate.txt
pop_up_admin.php
SHOP_ADMINISTRATION.php
SHOP_ADMINISTRATION_ARTIKEL.php
SHOP_ADMINISTRATION_AUFRUF.php
SHOP_BACKUP.php
SHOP_BESTELLUNG.php
Shop_Einstellungen_Menu_1.php
Shop_Einstellungen_Menu_Kategorien.php
SHOP_KUNDE.php
SHOP_LAYOUT.php
SHOP_SETTINGS.php
SHOP_VERSANDKOSTEN.php

Top Frame und Welcome Page

content.php
top.php

Template Dateien

template_initialize.php
template_ADMIN_initialize.php
template_create.sql
template_del.sql
template_insert.sql
template_remove.pl

 © 2001 R. Glanzmann & J. Fontanil